TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

OLLI LINDSTRÖM

STANDARDOITU TIEDONKERUU TEOLLISESSA TUOTANTO-YMPÄRISTÖSSÄ

Diplomityö

Tarkastaja: professori Jarmo Harju
Tarkastaja ja aihe hyväksytty
Tieto- ja sähkötekniikan tiedekunta-neuvoston kokouksessa 3. joulukuuta 2014

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Signaalinkäsittelyn ja tietoliikennetekniikan koulutusohjelma
LINDSTRÖM, OLLI: Standardoitu tiedonkeruu teollisessa tuotantoympäristössä
Diplomityö, 61 sivua
Helmikuu 2015
Pääaine: Tietoliikenneverkot ja protokollat
Tarkastaja: professori Jarmo Harju
Avainsanat: automaattinen tiedonkeruu, tuotantoympäristö, kenttäväylä, järjestelmäintegraatio, ISA, MESA, OPC, UA, tietoturva

Tiedonkeruujärjestelmiä ollaan hyödynnetty tuotantoprosessien tuottaman datan keräämisessä jo pitkään. Nämä järjestelmät ovat kuitenkin tyypillisesti olleet erillisiä järjestelmiä, eikä dataa ole voitu kuljettaa hyödynnettäväksi muihin tietojärjestelmiin. Tiedonkeruun standardoinnin ja järjestelmäintegraation tavoitteena on, että tehtaan lattiatason järjestelmistä kerätty data saadaan kuljetettua aina liiketoimintatasolle asti, jolloin sitä on mahdollista hyödyntää useissa eri sovelluksissa. Tässä työssä perehdytään järjestelmäintegraatioon liittyviin standardeihin ja parhaisiin käytäntöihin, OPC-integraatiotekniikkaan sekä tietoturvamalleihin automaatioverkon suojaamiseksi.

Työ koostuu kirjallisuustutkimusosasta sekä osasta, joka käsittelee työn tilannutta yritystä ja sen lähtötietojen kartoittamista integraatiohanketta varten. Kirjallisuustutkimusosassa paneudutaan aluksi alan standardeihin ja tietomalleihin, joissa kuvataan teollisen tuotantoympäristön toimijoita sekä tieto- ja datavirtoja. Seuraavaksi käsitellään työn kannalta oleellisia automaatiojärjestelmiin liittyviä tiedonsiirtotekniikoita sekä Microsoftin kehittämää COM/DCOM-tekniikkaa, johon perinteinen OPC perustuu. Tämän jälkeen tutustutaan edellä mainittuun integraatiotekniikkaan eli OPC:iin, sen seuraajaan OPC UA:iin sekä tiedonkeruun tietoturvallisuusmenettelyihin.

Kohdeyritystä käsittelevässä osuudessa kerrotaan kohdeyrityksen lähtötilanteesta ja millaisia ongelmia ja tarpeita tiedonkeruuhankkeella tulisi pystyä ratkaisemaan. Osiossa käsitellään myös kohdeyritykselle suunniteltua tietomallia ja tuotantolinjoista tehtyä dokumentaatiota. Työn aikana tehdyt havainnot ja havaintojen pohjalta tehdyt suositukset jatkotoimenpiteistä on esitetty osion loppupuolella.

Työn tuloksena saatiin kartoitettua kohdeyrityksen lähtötilanne tiedonkeruuhankkeen tarpeiden ja tuotantolinjojen osalta. Havaintojen perusteella pystyttiin muodostamaan kokonaiskuva jatkotoimenpiteistä, joiden avulla tiedonkeruuhanketta voidaan viedä eteenpäin. Ohessa laadittiin tiedonkeruuhankkeen kannalta ohjeistus automaatiojärjestelmien hankintaa varten, jotta jatkossa oleelliset asiat osataan ottaa huomioon. Tällöin järjestelmien käyttöönotto nopeutuu ja järjestelmät ovat yhtenäisempiä.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Signal Processing and Communications Engineering
LINDSTRÖM, OLLI: Standardized Collection of Production Data in Factory Environment
Master of Science Thesis, 61 pages
February 2015
Major: Communication Networks and Protocols
Examiner: Professor Jarmo Harju
Keywords: automated data collection, manufacturing environment, fieldbus, system integration, ISA, MESA, OPC, UA, security

Data collection systems have been in use for long in order to collect data from manufacturing processes. However, these systems have typically been individual, separated systems resulting in a situation where collected data can't be fully utilized. The main goal for standardizing data collection and system integration is to enable data to be transferred from factory floor level devices to enterprise level applications. This thesis studies standards and best practices related to system integration, OPC integration technology and security models for secure automation environment are inspected.

This thesis consists of two main parts. The first part is a theoretical study of data collection and system integration. The second part studies the company that ordered this work and the mapping of starting points for integration project in the company. In the first chapter of the literal part standards and information models that describe industrial manufacturing environment, actors and data flows are examined. Next chapter inspects technologies for data transferring used in automation systems and COM/DCOM-technology developed by Microsoft. In the end, OPC and its successor OPC UA and security aspect in automation networks are examined.

In the part that studies the company, all problems and demands related to current information systems and things that are pointed out for being solved during integration process are presented. This part covers the information model that was designed for target company and the documentation about mapping of company's production lines. Observations and recommendations for future improvements are presented in the end of this part.

As a result of this work, the target company's current starting point for implementing integrated data collection systems was clarified. Based on the findings, overall picture of integration process and a list of things to be done before moving further in this project were discussed. Concerning data collection, instructions for acquisition of automation systems were also drafted to achieve homogenous systems in the future and to avoid mistakes made in the past.

ALKUSANAT

Tämä diplomityö on tehty Suomen mittakaavassa suurelle teolliselle tuotantoyritykselle, jota ei työn aiheen takia haluttu tuoda julki. Työ tehtiin pääosin syksyn 2014 aikana. Tampereen teknillisen yliopiston puolesta työn tarkastajana on toiminut professori Jarmo Harju.

Haluan kiittää kohdeyritystä mielenkiintoisesta ja monipuolisesta diplomityöaiheesta, sekä kohdeyrityksen työn ohjaajaa ja työtäni edesauttaneita henkilöitä. Lisäksi haluan kiittää tietenkin työn tarkastajaa Jarmo Harjua hyvistä työhön liittyvistä neuvoista ja kommenteista, sekä laadukkaasta opetuksesta opintojeni aikana.

Haluan kiittää myös perhettäni, avopuolisoani ja ystäviäni, jotka ovat jaksaneet kannustaa ja joiden ansiosta opiskeluaika on ollut mielekästä.

Tampereella 13. helmikuuta 2015

Olli Lindström

SISÄLLYS

TERMIT JA NIIDEN MÄÄRITELMÄT

A&E	Alarms and Events, OPC:n spesifikaatio, joka tiedottaa prosessissa ilmenneistä tapahtumista ja hälytyksistä.
CAN	Control Area Network, väylätekniikka joka on kehitetty sulautettuja järjestelmiä varten.
COM	Component Object Model, Microsoftin kehittämä ohjelmistokomponenttien tekemiseen tarkoitettu oliomenetelmä.
Control Domain	kontrollointialue, joka sisältää kaikki tuotannon hallintaan liittyvät toiminnot.
DA	Data Access, OPC:n spesifikaatio, jonka käyttötarkoitus on reaaliaikaisen datan siirtäminen.
DCOM	Distributed Component Object Model, COM:n seuraaja, jota voidaan käyttää hajautetusssa verkkoympäristössä.
DMZ	Demilitarized Zone, aliverkkotyyppi, jota käytetään erottamaan verkkoja, joiden tietoturvatasot poikkeavat merkittävästi toisistaan.
ERP	Enterprise Resource Planning, toiminnanohjausjärjestelmä.
HDA	Historical Data Access, OPC:n spesifikaatio, jonka avulla kerättyä dataa voidaan käyttää myös myöhemmin.
IDS	Intrusion Detection System, tunkeutumisen tunnistamisjärjestelmä.
ISA	Instrumentation, Systems, and Automation Society, maailmainlaajuinen, ei-kaupallinen organisaatio, joka standardoi tehdasautomaatioratkaisuja ja kouluttaa ihmisiä.
ISA-95 ja ISA-88	toimintamalleja, jotka on kehitetty ohjaamaan järjestelmäintegraation toteuttamista teollisuudessa.
Istunto	Kahden osapuolen välinen muodostettu kommunikointiyhteys

Käyttäjävarmenne	identifioi käyttäjän, joka yrittää muodostaa yhteyden palvelimelle.
MES	Manufacturing Execution System, tuotannonohjausjärjestelmä.
MESA	Manufacturing Enterprise Solutions Association, maailmanlaajuinen, voittoa tavoittelematon yhteisö, joka kehittää ja jakaa alan parhaita käytäntöjä.
Modbus	Modiconin kehittämä kenttäväyläprotokolla.
Ohjelmistovarmenne	identifioi OPC UA -tuotteen version. Sisältää lisäksi tiedon version tukemista profiileista.
OLE	Object Linking and Embedding, Microsoftin kehittämä tekniikka, joka mahdollistaa olioiden linkittämisen ja sulauttamisen toisiin olioihin.
OPC	OLE for Process Control, standardi, jonka tehtävä on taata yhteensopivat rajapinnat ja sujuva tiedonsiirto eri toimittajien automaatiojärjestelmien ja -laitteiden välillä.
PCS	Process Control Systems, prosessinohjausjärjestelmät.
PKI	Public Key Infrastructure, julkisen avaimen järjestelmä, käytetään varmenteiden hallinnointiin.
Pino	OPC UA -sovellusten yhteinen osa, joka kattaa matalan tason toiminnallisuuden.
Profibus	Siemensin kehittämä kenttäväyläprotokolla.
Proxy	välityspalvelin tai välityskomponentti.
RTE	Real-Time Ethernet, Ethernetiä hyödyntävä kenttäväylätekniikka.
UA	Unified Architecture, OPC:n uusin standardi, joka määrittää abstraktin palvelukeskeisen arkkitehtuurin.

USB	Universal Serial Bus, hyvin yleisesti käytössä oleva sarja-väylätekniikka
SDK-kerros	Software Development Kit, kattaa OPC UA -sovellusten korkean tason toiminnot.
Sovellusvarmenne	identifioi isäntäkoneella käynnissä olevan OPC UA -sovelluksen instanssin.
Varmenne	digitaalinen dokumentti, jolla voidaan todentaan haluttu asia.
Wrapper	konvertoi OPC:n komponentit OPC UA:n komponentteja vastaavaksi tai päinvastoin.

1 JOHDANTO

Automaatiojärjestelmiä on käytetty jo kymmeniä vuosia teollisissa tuotantoympäristöissä, ja niistä on kerätty arvokasta dataa tiedonkeruujärjestelmien avulla. Suurelta osin tiedonkeruujärjestelmät ovat olleet erillisiä järjestelmiä ja datan kerääminen ei ole ollut kaikissa tapauksissa automatisoitua, minkä takia tiedonkeruujärjestelmien ja datan täyttä potentiaalia ei ole voitu täysin hyödyntää. Lisäksi automaatiojärjestelmien elinkaari on hyvin pitkä ja samassa tuotantoympäristössä voi olla järjestelmiä eri vuosikymmeniltä, mikä aiheuttaa haasteita tiedonkeruun kannalta. Tuotannon ja liiketoiminnan tietojärjestelmien kokonaisvaltaisella järjestelmäintegraatiolla on kuitenkin mahdollista saavuttaa hyvin suuria taloudellisia ja laadullisia hyötyjä sekä vähentää manuaalisen työn määrää, minkä takia kyseinen hanke kiinnostaa työn tilannutta yritystä.

Työn aihe saatiin Suomen mittakaavassa isolta teolliselta tuotantoyritykseltä, joka työllistää satoja työntekijöitä. Työ oli luonteeltaan kartoitustyö, jossa pyrittiin muodostamaan kokonaiskuva järjestelmäintegraatiosta sekä kartoittamaan ja päivittämään kohdeyrityksen lähtötiedot hanketta varten. Työ on siinä mielessä ajankohtainen, että viime aikoina järjestelmäintegraatiosta ja sen tarpeesta puhutaan koko ajan enenevässä määrin. Myös Suomessa isot teolliset tuotantoyhtiöt alkavat hiljalleen havahtua siihen kuinka suuresta ja tärkeästä asiasta on kyse.

1.1 Työn tavoitteet ja rajaus

Työn tilannut yritys asetti työlle kolme tavoitetta. Ensimmäinen tavoite oli kartoittaa yrityksen ja tuotantolinjojen nykytila, jotta jatkotoimenpiteitä voitaisiin suunnitella tiedonkeruuhanketta varten. Toinen tavoite oli luoda tiedonkeruulle standardin mukainen tietomalli, josta ilmenee mitä dataa tuotantoprosesseista halutaan kerätä, sekä miten ja mihin dataa siirretään. Kolmantena tavoitteena oli tarkastella, miten tuotantolinjojen yhdistäminen toimistoverkkoihin toteutetaan tietoturvallisesti ja mitä tietoturvamekanismeja tulisi käyttää.

Tilannekartoitus rajattiin käsittelemään yrityksen tiettyjä tehtaita ja tehtaiden tuotantolinjoja. Alunperin tarkoituksena oli myös tarkemmin tutkia, mitä dataa kyseisistä tuotantolinjoista on mahdollista kerätä, mutta tämä osoittautui työn aikana mahdottomaksi. Vaikka suurimmat hyödyt saavutetaan kokonaisvaltaisella järjestelmäintegraatiolla, niin tietomallin osalta työ rajattiin käsittelemään ISA-95:n hierarkiamallin mukaisesti tuotannonohjauskerroksen ja sen alapuolisten kerrosten tieto- ja datavirtoja. Myöskään taloudellisia vaikutuksia ei sen enempää tässä työssä käsitellä. Tiedonkeruun tietoturvallisuutta ja tietoturvamenettelyjä tarkasteltiin yleisesti, eikä kohdeyrityksen verkkoinfrastruktuuria tai muita verkkoratkaisuita otettu huomioon.

1.2 Työn rakenne

Työn luonteesta johtuen työ pohjautuu hyvin pitkälti teoriaan ja erilaisiin standardeihin, eikä työssä ole toteutus- tai testausvaiheita. Ensiksi esitellään tietomallin suunnittelun kannalta oleellisimmat organisaatiot, standardit ja standardeista johdetut tietomallit, joihin on sovellettu parhaita käytäntöjä. Tietomalleissa määritetään tuotantoympäristön toiminnot ja toimintojen väliset tietovirrat, jotta tuotannosta kerättyä dataa ja muista tietojärjestelmistä tuotua tietoa voidaan yhdistää tuotantoprosessien seuraamista varten. Seuraavaksi käsitellään työn kannalta oleellisia tekniikoita, joita yleisesti käytetään automaatiojärjestelmissä ja joita on mahdollista hyödyntää tiedonkeruussa. Varsinaiset integraatiotekniikat eli OPC ja OPC UA käsitellään luvuissa 4 ja 5. Myös muita integraatiomenetelmiä on olemassa, mutta jo työn alussa oli selvää, että OPC-tekniikoita tullaan hankkeessa käyttämään. Viimeisissä teoriakappaleissa tarkastellaan, miten tietoturvallisuus tulee ottaa huomioon, kun automaatioverkot yhdistetään yrityksen toimistoverkkoon, ja minkälaisia toimenpiteitä on mahdollista suorittaa verkkojen suojaamiseksi. Samalla esitetään OPC UA:n sisältämät tietoturvaratkaisut.

Kohdeyritystä käsittelevä osuus esitetään luvuissa 8 ja 9. Luvussa 8 on kooste yrityksen työntekijöiden haastatteluista, joiden perusteella kartoitettiin nykyisten tiedonkeruujärjestelmien ja muiden tietojärjestelmien ongelmakohdat, sekä millaisia tarpeita tiedonkeruuhankkeella tulisi ratkaista. Samalla käydään läpi miten tiedonkeruun tietomalli suunniteltiin ja millaisia haasteita suunnittelun aikana kohdattiin. Luvussa 9 on esitetty tuotantolinjojen laitekartoitus sekä laitekartoituksen aikana tehdyt havainnot. Molempien lukujen loppuosassa on annettu suositukset tehtyjen havaintojen perusteella jatkoa ajatellen.

2 JÄRJESTELMÄINTEGRAATIO JA TIETOMALLIN SUUNNITTELU

Yksi työlle asetetuista tavoitteista oli suunnitella kohdeyritykselle standardi tietomalli tiedonkeruuta varten, jotta tiedonkeruuta saataisiin yhtenäistettyä. Tietomallissa on tarkoitus määrittää mitä tietoa on saatavilla ja mahdollista kerätä, miten ja missä muodossa tieto tallennetaan, sekä miten tietoa ohjataan tuotannon tietojärjestelmiin. Tietomallin suunnittelussa lähtökohtana oli luonnollista käyttää jo olemassa olevia parhaita käytäntöjä ja standardeja. Tässä luvussa käsitellään työn kannalta oleellisimmat organisaatiot, standardit ja tietomallit, joiden pohjalta kohdeyrityksen tietomallia lähdettiin suunnittelemaan.

2.1 ISA, ISA-standardit ja MESA

ISA (Instrumentation, Systems, and Automation Society) on maailmanlaajuinen, eikaupallinen organisaatio, joka määrittelee tärkeimmiksi tehtävikseen tehdasautomaatioon liittyvät standardoinnit, sertifioinnit, julkaisut sekä konferenssit ja koulutukset. ISA on julkaissut monia tunnettuja automaatiostandardeja, kuten ISA-88 (Batch Control) ja ISA-95 (Enterprise-Control System Integration), jotka ovat tämän työn kannalta keskeisimmät ISA-standardit.

ISA-95 on toimintamalli, joka on kehitetty ohjaamaan järjestelmäintegraation toteuttamista teollisuudessa. ISA-95:n tavoitteena on vähentää yritys- ja automaatiojärjestelmien integrointiin liittyviä riskejä, virheitä ja kuluja sekä yksinkertaistaa uusien järjestelmien toteutusta niin, että ne ovat helposti integroitavissa ja yhteensopivia muiden järjestelmien kanssa. Standardi koostuu terminologiasta sekä useista malleista, joiden tehtävänä on edesauttaa eri ammattikuntien välistä yhteystyötä sekä kuvata järjestelmäintegraation ongelmia eri perspektiiveistä. Myös järjestelmien välillä kulkeva tieto on standardissa määritelty, mikä on johtanut ERP- ja MES-järjestelmäkehittäjien luopumaan omista rajapintamäärittelyistä ja ottamaan käyttöön ISA-95:n määrittämät rajapinnat.

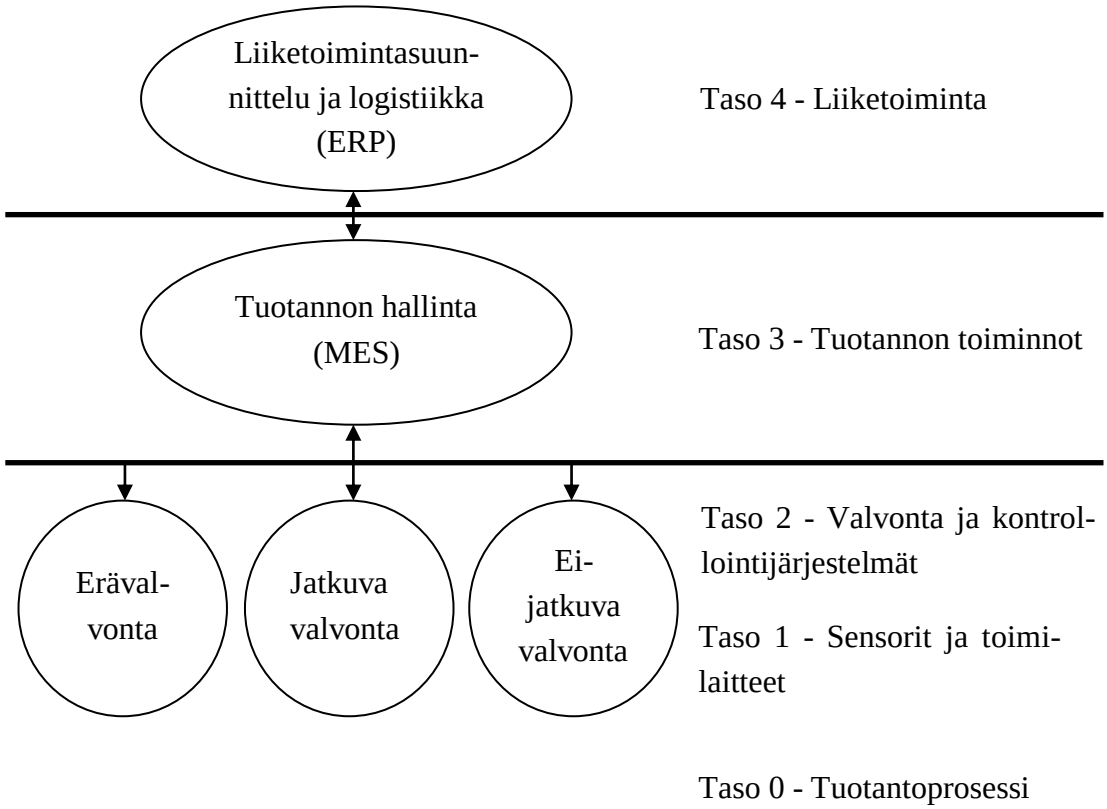
ISA-88 standardi sisältää niin ikään terminologiaosuuden sekä teolliseen erätuotantoon liittyvät standardit mallit ja datarakenteet. Standardi koostuu neljästä osasta, joista osa 4: Batch Production Records on keskeinen tämän työn kannalta, sillä se kuvaa, miten tuotantoerää koskevia tietoja tulisi käsitellä. Kuvaus kattaa mallit ja määritelmät tiedon tallentamiselle ja tiedon vaihdolle tuotantojärjestelmien välillä.

MESA (Manufacturing Enterprise Solutions Association) on maailmanlaajuinen, voittoa tavoittelematon yhteisö, jonka jäseninä toimii suuria laitevalmistajia, järjestel-

mätoimittajia, konsultteja, analyytikkoja ja akateemisia toimijoita. Suuren ja monipuolisen yhteisön tavoitteena on kehittää optimoituja, tietoteknisiä sovelluksia liiketoiminnan ja tuotannon tarpeisiin, sekä jakaa alan tietoa ja parhaita käytäntöjä.

2.2 ISA-95:n toiminnallinen hierarkiamalli

Alla olevassa kuvassa 1 on esitetty yksinkertaistettu ISA-95:n määrittämä toiminnallinen hierarkkiamalli.



Kuva 1: ISA-95:n määrittämä toiminnallinen hierarkiamalli.

Kuvan 1 mukaisesti ISA on erotellut tuotteita valmistavan organisaation toiminnallisuudet eri tasoihin. ISA-95-standardi käsittelee tasojen 3 ja 4 välistä integraatiota ja ISA-88 tasojen 2 ja 1 välistä kontrollointiprosessia. Standardit ovat kuitenkin osittain päällekkäisiä ja niiden välinen raja häilyvä.

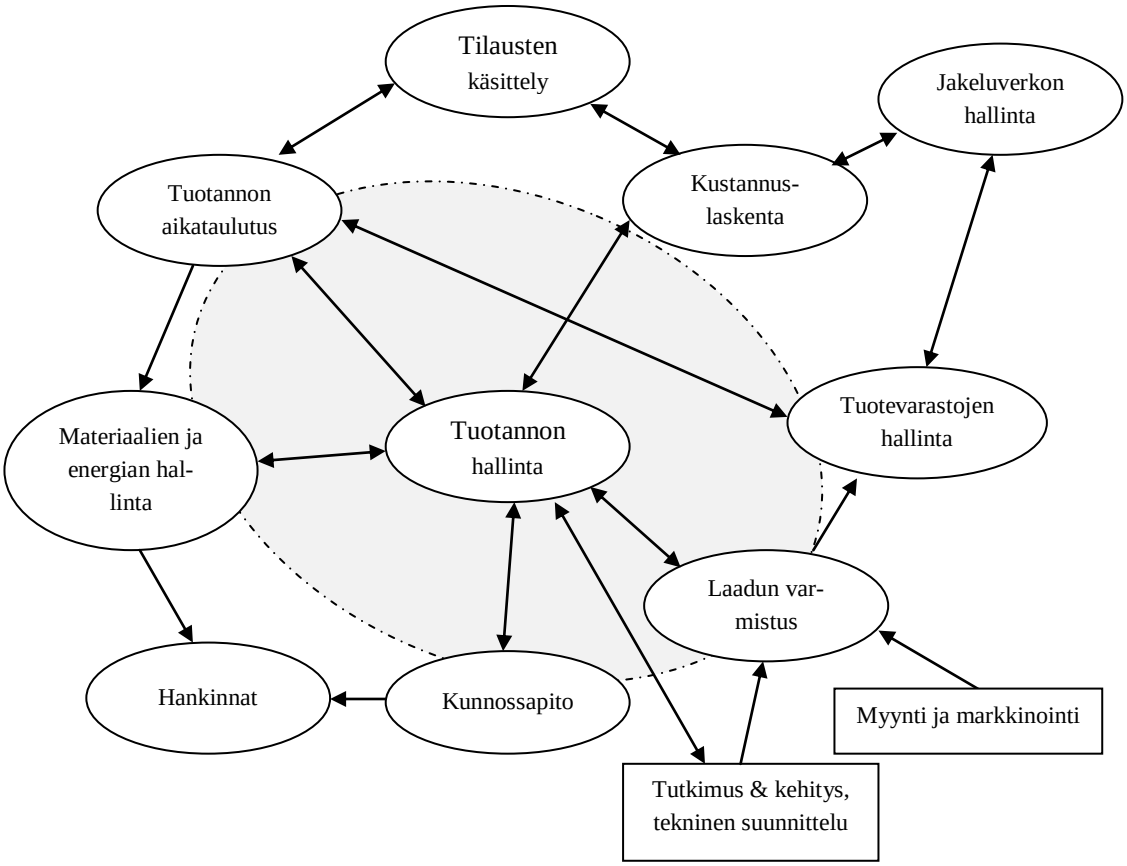
Taso 4 käsittää liiketoimintasuunnittelun ja logistiikan. Tästä tasosta käytetään järjestelmämielessä yleisesti nimitystä ERP-kerros. ERP-kerroksella vahvistetaan tehtaan perusaikataulut muun muassa tuotannon, materiaalikulutuksen ja toimituksien suhteen. Toiminnan aikaväli vaihtelee vuosista päiviin ja toiminta ulottuu tehtaan ulkopuolelle. Tasosta 3 käytetään yleisesti nimitystä MES-kerros. MES-kerroksen toimintoihin kuuluu työn ja reseptiikan kontrollointi halutun tuotteen valmistamiseksi sekä tuotanto-

prosessin optimointi ja tuotantokirjauksien ylläpito. Aikavälin vaihtelu on muutamista päivistä sekunteihin.

Tasoista 2 ja 1 käytetään yhteisesti nimitystä PCS-kerros, jolla tarkoitetaan prosessin ohjauskerrosta. Tätä nimitystä harvemmin kuitenkaan näkee suomalaisessa kirjallisuudessa käytettävän. PCS-kerroksen toiminnan aikaväli voi vaihdella muutamista tunneista sekunnin murto-osiin. Taso 0 kuvaa itse tuotantoprosessia. Tasot 3-0 muodostavat kokonaisuudessaan ISA:n määrittämän tuotannon kontrollointialueen (Control Domain), joka käsittää tehtaan sisällä tapahtuvat toiminnot. [1] Kohdeyrityksen tietomalli rajattiin koskemaan vain kontrollointialuetta eli liiketoimintataso rajattiin työn ulkopuolelle.

2.3 Tärkeimmät tietovirrat tuotantoympäristössä

ISA-95:n liiketoiminta-kontrollointialue -mallissa määritetään tuotannon hallinnan kannalta keskeiset toiminnot ja tietovirrat toimintojen välillä. Kuvan 2 mukaisesti harmaan alueen sisäpuolelle jäävät toiminnot ja tietovirrat kuuluvat tuotannon kontrollointialueeseen. Toiminnoista saatetaan käyttää eri nimityksiä eri organisaatioissa ja niiden väliset suhteet ja rajat saattavat poiketa esitetystä mallista. Lisäksi toimintojen tärkeys ja kompleksisuus ovat organisaatio- ja toimialakohtaisia. [1]



Kuva 2: Tuotannon tietovirrat

Kontrollointialueen ulkopuolelle jää hankinnat, tutkimus ja kehitys, tekninen suunnittelu, myynti ja markkinointi, tilausten käsittely, kustannuslaskenta sekä jakeluverkon hallinta. Myynti ja markkinointi, sekä tutkimus ja kehitys ovat lisäksi tyypillisesti omia yksiköitään. Nämä toiminnot kuuluvat liiketoimintatasolle, eikä niitä näin ollen oteta huomioon. Taulukkoon 1 on listattu toimijoiden väliset tärkeimmät tietovirrat, joidenka suunnat ovat nuolten mukaisia.

Taulukko 1: Tärkeimmät tietovirrat tuotannon hallinnassa.

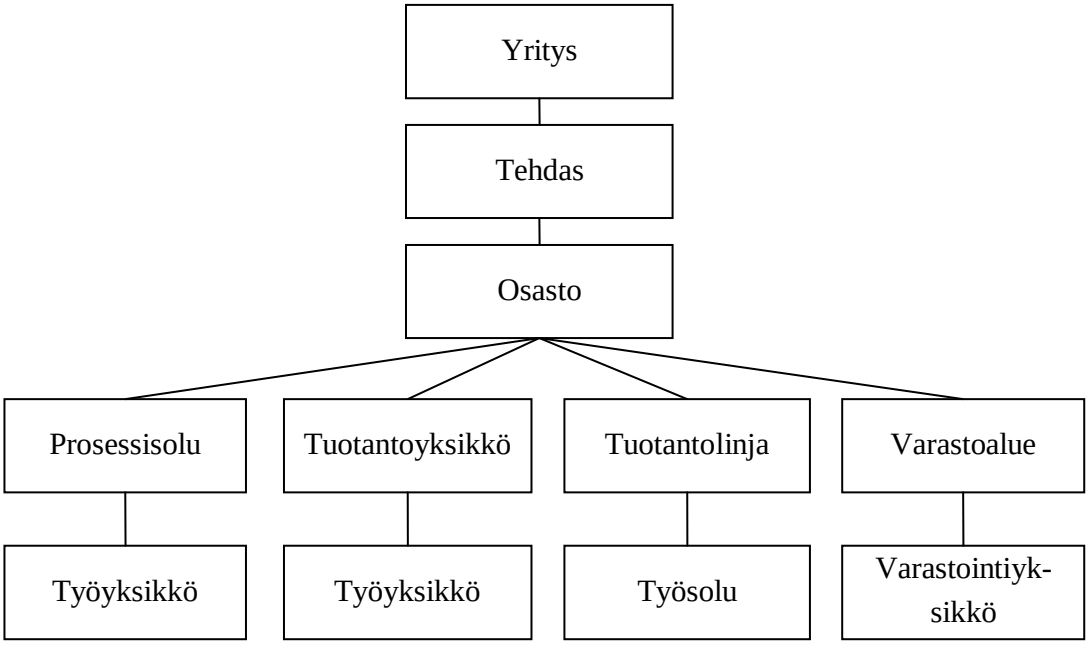
Tuotevarastojen hallinta	→ Prosessidata	Tuotannon hallinta
Tuotannon aikataulutus	→ Tuotantokapasiteetti → Tuotannon suunnitelma ← Aikataulut	
Laadun varmistus	→ Prosessidata ← Laatustandardit ja asiakasvaatimukset ← Laadunvarmistus tulokset	
Materiaalien- ja energian hallinta	→ Lyhyen aikavälin materiaali- ja energiavaatimukset ← Materiaali- ja energiavarastot	
Kunnossapito	→ Kunnossapitopyynnöt → Kunnossapitostandardit ja käytännöt ← Kunnossapidon tekninen palaute ← Kunnossapitovastaukset	
Kustannuslaskenta	→ Tuotannon kulut ja tehokkuus ← Tuotannon kulutavoitteet	
Tuotannon aikataulutus	→ Pakkausaikataulut ← Valmistuneiden tuotteiden varasto	Tuotevarastojen hallinta

2.4 Oliomallit tietovirtojen muodostamiseksi

Edellä mainitut tietovirrat saadaan koottua kolmesta MESA:n määrittämästä oliomallista, joiden perustana on käytetty ISA-95 ja ISA-88 -standardeja: välinemallista (Equipment), materiaalimallista (Material), henkilömallista (Personnel) (yksilö/työntekijä) sekä prosessisegmenttimallista (Process Segment). [1]

Välinemalli koostuu kolmesta osasta. Yksi osa sisältää tiedon itse välineestä, kuten määritelmän välineestä, välineen ominaisuuksista ja sitä koskevista kunnossapitopyynnöistä ja -määräyksistä. On myös mahdollista, että välineen tuotantokapasiteetti on ilmoitettu näissä tiedoissa. Toinen osa kuvaa välineen luokkaa ja sen tietoja, kuten minkälaisia ominaisuuksia luokan välineillä on. Kolmas osa sisältää tiedon testauksesta, kuten vastaavatko välineen ominaisuudet ja kapasiteetti ilmoitettuja arvoja.

Välinemalli on hierarkkinen, mikä tarkoittaa sitä, että väline voi tässä tapauksessa tarkoittaa mitä tahansa tuotannollista yksikköä. Esimerkiksi tehdas on väline siinä missä yksittäinen työsolukin. Välineiden lukumääräsuhteita ei ole rajoitettu vaan yksiköt voivat koostua useammista muista yksiköistä kuvassa 3 olevan mallin mukaisesti. [2]



Kuva 3: Hierarkkinen välinemalli [2]

Prosessisolu, tuotantoyksikkö, tuotantolinja ja varastoalue ovat kaikki työkeskittymiä, jotka koostuvat pienemmistä yksiköistä, kuten työyksiköistä tai -soluista.

Materiaalimallissa kuvataan materiaali, sen ominaisuudet ja mihin erään materiaali kuuluu. Materiaali voi olla raakamateriaa, puolivalmiste tai lopullinen tuote. Erätiedoissa kuvataan saatavilla olevan materiaalin määrä, materiaalin tilan sekä materiaalin sijainti. Lisäksi materiaalit luokitellaan niiden määritelmien perusteella ryhmiksi eri tuotantoprosesseja varten. Materiaalien ominaisuuksien testaamista ja todentamista varten on osa, joka sisältää tiedot laadunvalvonnan testeistä. Testitiedoissa määritetään muun muassa testispesifikaation ja listan testatuista ominaisuuksista. [3]

Henkilömalli sisältää muiden mallien tapaan henkilön tiedot ja luokat, joihin henkilöt on luokiteltu. Malli sisältää tiedot henkilöiden soveltuvuustestistä, jolla

varmistetaan, että henkilö on koulutukseltaan ja kokemukseltaan soveltuva tiettyyn työtehtävään. [4]

Prosessisegmenttimalli loogisesti yhdistää resurssit eli välineet, materiaalit ja henkilöt tuotantoprosessin tarpeisiin. Mallissa määritetään mitä resursseja tarvitaan, resurssien määrät ja mahdolliset erityisresurssit, kuten esimerkiksi erityisen tuotantovälineen. Prosessisegmenttimallit ovat organisaatio- ja prosessikohtaisia, eivätkä ne aina liity tuotantoon. Ainakin kolme yleistä prosessisegmenttimallia on olemassa. Tuotantosegmentti ja sen sisältämät toiminnot liittyvät siihen, miten raakamateriaalit muuntautuvat keskeneräisiksi tuotteiksi tai miten keskeneräiset tuotteet muuntautuvat valmiiksi tuotteiksi. Siirtosegmentti sisältää toiminnot, jotka liittyvät materiaalien liikuttamiseen ja seuraamiseen. Valvontasegmentissä testataan ja vahvistetaan materiaalien laatu ja sopivuus. [5]

2.5 Ajoerätietojen muodostaminen

Ajoerätietojen muodostaminen on kuvattu MESAn Batch Production Record -mallissa ([6]). Ajoerätiedot käsittävät kaikki toiminnot ja tapahtumat, mitä erän tuottamiseen ja hallintaan liittyy ja niiden perusteella voidaan siis jäljittää koko prosessiketju, jonka läpi valmis tuote on kulkenut.

Ajoerätietoja muodostettaessa joudutaan käyttämään ja yhdistämään lähes kaikkia MESA:n oliomalleja, minkä takia ajoerätietojen rakennetta ei tässä kokonaisuudessaan käydä läpi. Ajoerätiedot sisältävät esimerkiksi seuraavia tietoja:

- erätunnisteen, jolla erä on mahdollista identifioida yksiselitteisesti
- erän valmistuspäivämäärä- ja muut pakkaus- tai tuotemerkinnät
- käytetyt raaka- tai puolivalmistemateriaalit, joita on käytetty erän valmistamiseen
- kaikki laitteet ja tuotantolinjat, joita on käytetty erän valmistamiseen
- laadunvalvontaan liittyvät mittaukset
- työntekijät, jotka ovat olleet fyysisesti erän kanssa tekemisissä ja osallistuneet erän valmistamiseen.

Kuten sanottu, malli on todella suuri, eikä sitä kaikilla toimialoilla kannata hyödyntää sellaisenaan, koska eri toimialoilla ajoerään liittyvien tietojen tärkeys vaihtelee ja kaikki tiedot eivät ole pakollisia tai niitä ei tarvita. Toisaalta esimerkiksi laki saattaa velvoittaa joillakin toimialoilla, että ajoerätietoja tallennetaan ja ne säilytetään laadunvalvonnallisista syistä.

3 TAUSTATIETOA KÄYTETYISTÄ TEKNIIKOISTA

Automaatiojärjestelmät hyödyntävät useita erilaisia protokollia ja tiedonsiirtotekniikoita, kuten sarja- ja kenttäväylätekniikoita. Tässä luvussa käydään läpi yleisimmät tekniikat sekä kerrotaan lyhyesti Microsoftin kehittämistä COM ja DCOM -tekniikoista, joihin perinteinen OPC perustuu.

3.1 Microsoft COM ja DCOM

Microsoft COM (Component Object Model) on olioihin perustuva menetelmä, jota käytetään keskenään vuorovaikutuksessa olevien ohjelmistokomponenttien tekemiseen. COM ei ole olio-ohjelmointikieli vaan standardi, joka määrittää oliomallin, sekä vaatimukset, joiden perusteella COM-komponentit voivat olla vuorovaikutuksessa muiden olioiden kanssa. COM on alustariippumaton eikä siis rajoita olioiden toteutusta millään muulla tavalla, minkä takia siitä käytetään binääristandardi-nimitystä eli standardi, joka astuu voimaan vasta, kun ohjelmakoodi on käännetty binäärikoodiksi. COM toimii pohjana muun muassa Microsoftin kehittämälle OLE-teknologialle (Object Linking and Embedding), jolla tarkoitetaan useiden eri sovellusten olioiden yhdistämiseen käytettyä tekniikkaa. [7]

DCOM (Distributed Component Object Model) on COM:n seuraaja ja siinä on otettu huomioon ohjelmien ja olioiden hajaantuminen verkkoympäristössä. DCOM käsittää joukon konsepteja ja rajapintoja, joiden avulla asiakassovellus voi pyytää palveluita olioilta, jotka sijaitsevat verkon muilla tietokoneilla. [8]

3.2 RS232, RS422 ja RS485

RS232, RS422 ja RS485 -standardit on kehitetty Electronic Industries Association (EIA) -yhdistyksen toimesta, minkä takia niistä käytetään myös nimityksiä EIA232, EIA422 ja EIA485. Standardeista ensimmäinen, RS232, kehitettiin 1960-luvulla korvaamaan puhelinkaapelit ja modeemit digitaalisessa datan siirrossa kahden tietokoneen tai muun päätelaitteen välillä. RS232-standardin mukaisessa tiedonsiirrossa käytetyt signaalit eivät ole balansoituja, minkä takia se sopii hitaaseen, lyhyen kantaman tiedonsiirtoon. Tiedonsiirto on rajoitettu myös ainoastaan kahden laitteen välille.

RS422-standardissa pyrittiin korjaamaan RS232-standardin rajoituksia. Tiedonsiirron signaalit balansoitiin, jolloin tiedonsiirtonopeudet nousivat ja kantama piteni.

Standardin myötä saatiin osittainen tuki useamman laitteen väliseen tiedonsiirtoon, mutta tiedonsiirto on hyvin rajoitettua. Tämän takia RS422-standardia ei suositella usean laitteen väliseen tiedonsiirtoon. RS485-standardi on RS422-standardin laajennos, joka korjasi tiedonsiirto-ongelmat usean laitteen välillä.

3.3 Kenttäväylä

Kenttäväyläteknologia on kehitetty 1970-1980 -luvuilla teollisten tietokoneverkkojen ja hajautettujen automaatiojärjestelmien tarpeisiin. Tiedonsiirto perustuu digitaaliseen, asynkroniseen sarjamuotoiseen kaksisuuntaiseen lähetykseen ja siirtomedioina voidaan käyttää kuparikaapelia, valokuitukaapelia tai radioaaltoja. Erilaiset siirtomediat mahdollistavat, että hyvin suurellekin alueelle hajautuneet kenttälaitteet on mahdollista yhdistää kontrollointijärjestelmään. Kenttäväylä on osoittautunut hyvin joustavaksi ja tehokkaaksi teknologiaksi ja siitä on kehitetty erilaisia variaatioita suurten järjestelmävalmistajien toimesta vastaamaan erilaisia teollisia vaatimuksia. IEC 61158 ja IEC 61784 standardit sisältävät kaikki merkittävimmät kenttäväyläteknologiat. [9]

3.3.1 Modbus

Modiconin kehittämä Modbus on ensimmäisiä teolliseen käyttöön suunnattuja kenttäväyläprotokollia. Modbus perustuu yksinkertaiseen master/slave -konseptiin, mikä on sen suurimpia vahvuuksia. Alunperin Modbus oli tarkoitettu vain Modiconin omaan käyttöön, mutta myöhemmin siitä tehtiin julkinen, avoin protokolla. Julkaisun jälkeen monet yritykset ottivat Modbusin käyttöönsä sen yksinkertaisuuden ja helppokäyttöisyyden vuoksi. Modbus-protokollasta on kolme versiota: Modbus ASCII, Modbus RTU ja Modbus TCP/IP.

Modbus ASCII on sarjaprotokolla ja ensimmäinen kehitetty Modbus-protokolla. Fyysisinä tiedonsiirtoportteina se tyypillisesti käyttää RS-232 tai RS-485 -sarjaväyliä. Modbus RTU -protokollan ainoa ero ASCII-protokollaan nähden on datan binaarikoodaus, jossa käytetään tavuja ASCII-merkkien sijaan, minkä takia RTU:n prosessointi- ja läpimenoaika on paljon pienempi. Uudemmissa järjestelmissä RTU-protokolla on hyvin pitkälti syrjäyttänyt ASCII-protokollan. Modbus TCP/IP -protokolla on uusin Modbus-protokolla ja se kehitettiin paljon ASCII ja RTU -protokollia myöhemmin. Modbus TCP/IP käytännössä kapsuloi RTU-paketin TCP/IP-paketin sisään, minkä takia sen käyttöönotto on yksinkertaista. TCP/IP-protokollan haittapuolena on, että se on muita teollisia Ethernet -protokollia hitaampi, jolloin sitä ei voida hyödyntää todella reaaliaikakriittisissä sovelluksissa, mutta valvontasovelluksissa se on silti hyvin käyttökelpoinen. [10]

Kuten mainittu, Modbus-protokollan toiminta on hyvin yksinkertaista. Modbus-master-laite kontrolloi kaikkea tiedonsiirtoa Modbus-väylässä. Master-laite lähettää väylään kiertokyselyn, johon slave-laitteet vastaavat omalla vuorollaan. Slave-laitteet voivat käytännössä olla huomaamattomia, koska slave-laitteet eivät millään tavalla ilmaise olemassaoloaan liityttyään väylään. Slave-laitteet eivät myöskään lähetä väylään

mitään ellei master-laite lähetä niille kyselyä. Modbus-protokollan heikkoutena on, ettei siihen ole suunniteltu mitään mekanismia vikatilanteita varten. Esimerkiksi slave-laite voi jatkaa virheellistä toimintaa ilmoittamatta siitä master-laitteelle. [11]

3.3.2 Profibus

Profibus kehitettiin 1990-luvulla ja Modbusiin verrattuna se on paljon modernimpi ja kehittyneempi protokolla. Modbusin tavoin se kuitenkin perustuu master-slave-konseptiin. Myös Profibus -protokollasta on kehitetty erilaisia variaatioita, kuten Profibus DP, Profibus PA, Profisafe, Profidrive ja Profinet, joista kenties merkittävimmät ovat Profibus DP ja Profinet.

Profibus DP (Decentralized Peripherals) on suunniteltu ensisijaisesti kenttätason laitteiden ja sovellusten väliseen, nopeutta vaativaan, tiedonsiirtoon. Tiedonsiirto on syklistä eli master-laite lähettää kiertokyselyn, johon slave-laitteet vastaavat vuorollaan. Profibusin tapauksessa laitteet kuitenkin käyvät läpi tietyn käynnistysprosessin liityttyään väylään. Myös kommunikaatiota on koordinoitu enemmän. Mikäli master-laite ei lähetä kyselyä slave-laitteelle, menee slave-laite turvatilaan (safestate). Tällöin master-laitteen tulee käydä käynnistysprosessi uudelleen, jotta tiedonsiirto voi jatkua. Lisäksi Profibusissa on ajastin, joka pitää huolta, että kaikki kommunikaatio tapahtuu tietyn aikavälin sisällä jokaisella syklillä. Modbusista poiketen, samassa Profibus-väylässä voi olla useita master-laitteita, jotka käyttävät niin sanottua token-mekanismia varmistaakseen, etteivät laitteet lähetä kiertokyselyä samaan aikaan. [9]

Siemens Point-Point Interface

Siemensin kehittämä Point-Point Interface (PPI) on master-slave-protokolla, joka pohjautuu Profibus-standardiin. Protokolla tukee linja- ja tähtimallisia verkkotopologioita. Dataa on mahdollista siirtää RS232 ja USB -siirtomedioiden kautta. [12]

Siemens Multi-Point Interface

Multi-Point Interface (MPI) mahdollistaa myös puumallisten verkkotopologioiden muodostamisen, jolloin verkko on mahdollista segmentoida useampaan osaan. Siirtomediana MPI käyttää RS485-standardia. [12]

3.3.3 Real-Time Ethernet

Real-Time Ethernet (RTE) -protokollat perustuvat kenttäväyläteknologiaan, mutta ne hyödyntävät Ethernetia OSI-mallin fyysisellä kerroksella, sekä siirtokerroksella. Ethernetin käytön myötä RTE-protokollat ovat kuitenkin paljon monipuolisempia ja niiden avulla verkkojen muodostaminen ja yhdistäminen on vaivattomampaa kuin muilla verkkotekniikoilla.

Profinet on yksi Real-Time Ethernet -protokollista. Profinet koostuu kahdesta mallista: Profinet CBA (Component-Based Automation) sekä Profinet IO. Profinet CBA on suunniteltu täyttämään automaatioteknologian vaatimukset modulaarisista, uudelleenkäytettävistä laitteista, sekä hajautetusta kontrolloinnista tehdasympäristössä. Profinet IO:ia käytetään ohjelmoitavien kontrollereiden ja muiden älykkäiden kenttätason laitteiden kanssa kommunikointiin. IO-mallin peruslähtökohtana on reaaliaikainen toiminnallisuus.

Profinet Migration Modelissa on määritetty proxyt, joiden avulla Profibus DP-protokolla voidaan kapseloida Ethernet-kehyksellä. Tällöin vanhemmat Profibus-laitteet voidaan liittää Ethernet-verkkoon Profinetin avulla ilman muutoksia alkuperäisiin laitteisiin. [9]

3.3.4 CAN

Control Area Network (CAN) -protokolla on sulautettuja järjestelmiä varten kehitetty väylätekniikka, jonka Bosch GmbH esitteli 1986. Vuonna 1993 protokollasta julkaistiin standardi ISO 11898.

Alunperin CAN-protokolla suunniteltiin ajoneuvojen ja kulkuvälineiden sisäiseen tiedonsiirtoon, mutta nykyään sitä käytetään hyvin laajasti erityyppisissä sovelluksissa, joissa tarvitaan reaaliaikaista tiedonsiirtoa, ja se on yksi merkittävimmistä väylätekniikoista. Myös teollisissa automaatiojärjestelmissä käytetään CAN-protokollaan pohjautuvia spesifikaatioita, kuten DeviceNet ja CANopen. [13]

3.4 USB

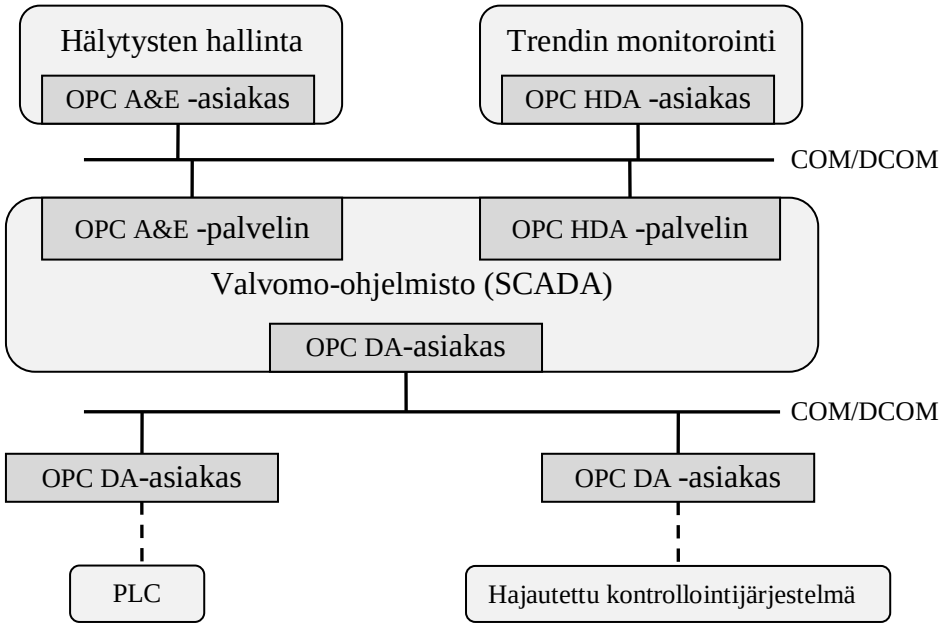
Universal Serial Bus (USB) on sarjaväylätekniikka, joka kehitettiin vuonna 1995 tietokoneen ja lisälaitteiden yhdistämiseen. Helppokäyttöisyyden, standardin liittimen, korkean suorituskyvyn ja alhaisten kustannusten vuoksi USB korvasi hyvin nopeasti vanhemmat liitäntätyypit tietokoneissa ja oheislaitteissa. USB:sta on tähän mennessä kehitetty neljä spesifikaatiota, USB 1.1, USB 2.0, USB 3.0 ja USB 3. USB:n kehitystyön aikana tiedonsiirtonopeudet ovat lähes tuhatkertaistuneet, virran hallinta on parantunut, minkä lisäksi USB:sta on kehitetty myös langaton variaatio. Kaikista parannuksista huolimatta USB on säilyttänyt pääsuunnittelutavoitteensa eli helppokäyttöisyyden ja yhteensopivuuden. [14]

Teollisissa automaatiojärjestelmissä USB-tekniikka on verrattain uutta, mutta sitä käytetään jossain määrin muun muassa ohjelmistopäivitysten yhteydessä. USB-tekniikka on kuitenkin näyttänyt potentiaalinsa muilla osa-alueilla, joten sen käyttö automaatiojärjestelmissä voi hyvinkin yleistyä tulevaisuudessa.

4 PERINTEINEN OPC

OPC on OPC Foundation -järjestön kehittämä ja ylläpitämä standardi, jonka tarkoituksena on taata yhteensopivat rajapinnat ja sujuva tiedonsiirto eri toimittajien automaatiojärjestelmien ja -laitteiden välillä. Alun perin OPC viittasi sanoihin OLE (Object Linking and Embedding) for Process Control, mutta OPC Foundation on antanut OPC:lle uuden merkityksen open connectivity via open standards, mikä kuvaa paremmin OPC:n nykyistä tarkoitusta. Luvun lähteenä on pääosin käytetty Lange:n ja Iwanitz:n julkaisemaa OPC - Openness, Productivity, and Connectivity -artikkelia ([15]).

OPC:n tiedonsiirto perustuu asiakas-palvelin-malliin, jossa automaatiojärjestelmän prosessidata on saatavilla OPC-palvelimen rajapintojen kautta. Perinteisen OPC:n rajapinnat pohjautuvat Microsoftin COM ja DCOM -teknologioihin, jolloin tiedonsiirto on Windows-käyttöjärjestelmäriippuvaista. Täyttääkseen erilaisten automaatiosovellusten vaatimukset perinteistä OPC-spesifikaatiota on laajennettu. Yhteisten osuuksien lisäksi se sisältää kolme tärkeää määritelmää, jotka käsitellään tässä työssä: OPC Data Access, OPC Alarms and Events (A&E) ja OPC Historical Data Access. OPC on saavuttanut de facto -standardin aseman käyttäjien ja kehittäjien keskuudessa ja suurin osa automaatiojärjestelmätoimittajista tarjoaa OPC-asiakas- ja palvelinrajapinnat tuotteilleen. OPC:n rajapintojen käytöstä on esitetty tyypillinen käyttötapaus kuvassa 4.



Kuva 4: Tyypillinen OPC-asiakkaiden ja -palvelinten käyttötapaus on toimia teollisuusautomaatiojärjestelmien välisinä rajapintoina. [16]

4.1 OPC Common Definitions and Interfaces

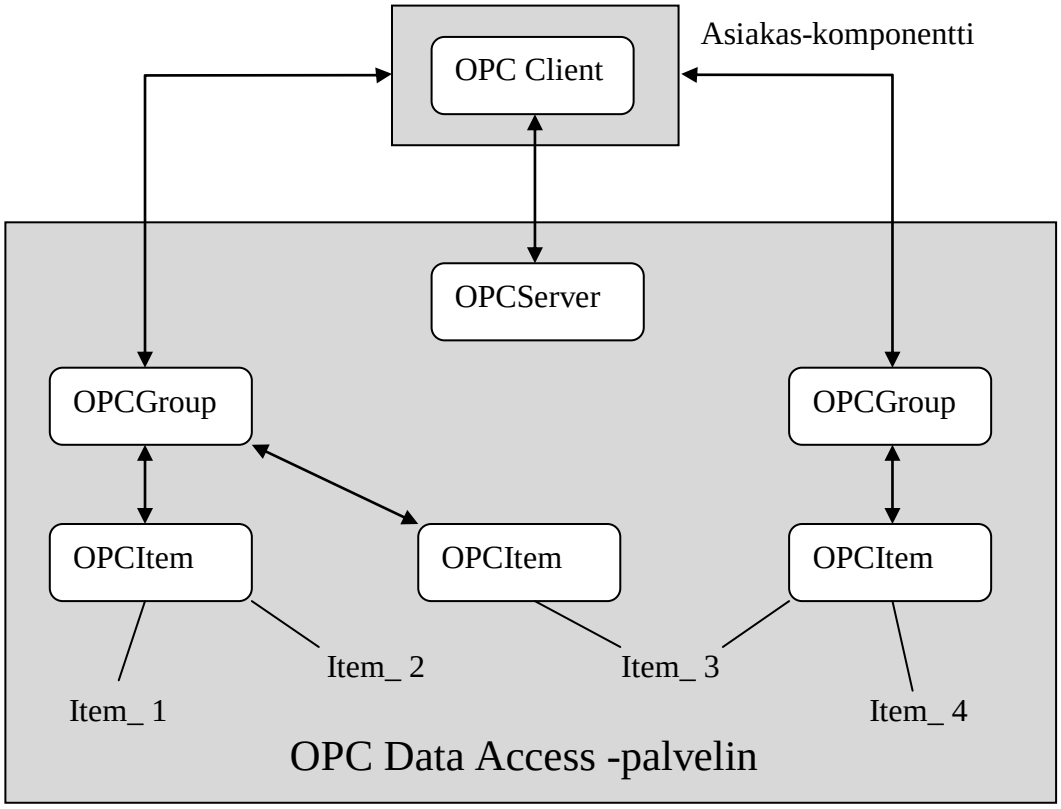
OPC Foundation huomasi valmistellessaan OPC Data Access -spesifikaatiota muiden spesifikaatioiden ohella, että osa määritelmistä oli relevantteja kaikille spesifikaatioille. OPC Common Definitions and Interfaces -spesifikaatioon on koottu kaikki määritelmät, jotka ovat yhteisiä muiden OPC-spesifikaatioiden kesken. Yhteisiä määritelmiä ovat:

- Kaikkien OPC-palvelimien pitää taata tietyt perustoiminnallisuudet, jotka on kuvattu pakollisina rajapintoina. Lisäksi palvelin voi tarjota valinnaisia rajapintoja.
- Asennusprosessi. Tietyn spesifikaation kaikki palvelimet ja asiakkaat käyttävät samoja komponentteja, jolloin niiden komponenttien on oltava saatavilla niissä tietokoneissa, joissa OPC-tuotteita käytetään.
- Palvelimen tunnistusprosessi. Rekisteritietokannassa olevat merkinnät sisältävät tarvittavan tiedon palvelimen tunnistamiseksi ja käynnistämiseksi. Seuraavat rekisterimerkinnät määrittelevät palvelimen:
    - Program Identifier (ProgId): Määrittää jokaisen DCOM-palvelimen. Käytetään erottamaan OPC-palvelimet muista DCOM-palvelimista.
    - Class Identification (CLSID): 128-bittinen, ainutlaatuinen tunniste, joka määrittää DCOM-palvelimen, joka on myös OPC-palvelin.
    - Application Identifier (AppId): Sisältää lisätietoa palvelimesta, kuten tietoturva-asetuksista. AppId ei saa olla sama kuin CLSID, vaikka se käytännössä onkin mahdollista.

4.2 OPC Data Access

OPC Data Access -rajapintojen yleisin käyttötarkoitus on reaaliaikaisen datan siirtäminen kenttätason ohjauslaitteilta näyttölaitteille tai käyttöjärjestelmille. OPC Data Access -spesifikaatiota pidetään tärkeimpänä määrittelynä OPC-spesifikaatioiden joukossa ja suuri osa muista määritelmistä on lisäyksiä tähän, koska OPC DA suurimmaksi osaksi täyttää OPC:n lupaukset yhteensopivuudesta ja liitettävyydestä. Tätä varten OPC DA -palvelimiin on toteutettu osoiteavaruus sekä toiminnallisuus, jolla osoiteavaruutta voidaan selata asiakkaan toimesta, jotta data olisi saatavilla mahdollisimman kätevästi.

Asiakas saa yhteyden palvelimeen muodostamalla yhteyden palvelinolioon (OPCServer), jolloin sillä on pääsy palvelimen osoiteavaruuteen (namespace). Palvelimen osoiteavaruus on identtinen kaikille asiakkaille, mutta asiakas voi luokitella osoiteavaruudessa olevat tietoalkiot (OPCItem) eri ryhmiin (OPCGroup), sekä määritellä tietoalkioiden hierarkian halutunlaiseksi. Tietoalkioiden kautta asiakas voi lukea, kirjoittaa ja valvoa prosessidataa valitsemalla halutun muuttujan palvelimelta. Kuvassa 5 on esitetty OPC DA -sovelluksen muodostavat komponentit, sekä OPC DA -palvelimen osoiteavaruus ja hierarkkinen rakenne.



Kuva 5: OPC DA -sovelluksen tiedonsiirrossa käytetyt komponentit, sekä komponenttien sisältämät oliot

Palvelimen käynnistyksen jälkeen asiakkaalla on pääsy OPCServer-olion rajapintaan. Rajapinnan toiminnallisuuteen kuuluu OPCGroup-olion eli ryhmien luominen, minkä mukaan muuttujien hierarkkinen rakenne määräytyy. OPCGroup-olion rajapintojen avulla asiakas voi pyytää palvelinta luomaan OPCItem-oliot ryhmän sisälle. OPCItem-oliot esittävät prosessidatan arvoja. Jotta arvoja voitaisiin lukea ja kirjoittaa tehokkaasti, niillä ei ole rajapintoja vaan toiminnallisuus löytyy OPCGroup-olion rajapinnoista.

Yhteensopivuuden takaamiseksi asiakkaan ja palvelimen välisen tiedonsiirron tulee olla standardimuotoista. OPC-spesifikaatio määrittää, että kaikki sovelluskohtaiset tietotyypit muutetaan DCOM-tietotyypeiksi, kun prosessidatan arvoja siirretään palvelimen ja asiakkaan välillä. Tietotyyppien muutos on sovelluskohtaista. OPC:n dataformaatti sisältää kuvauksen prosessidata-arvon tietotyypistä, tiedon laadusta sekä aikaleiman. Tietotyyppi kertoo minkälaisesta arvosta on kyse, kuten onko tieto numeerista vai sisältääkö se muitakin merkkejä. Kuvaus tiedon laadusta kertoo kuinka tarkkaa ja käyttökelpoista tieto on – hyvä, huono tai epävarma. Aikaleima kertoo, koska tieto on saatu, ja se luodaan palvelimella tai mahdollisesti jo automaatiolaitteella. Näillä tiedoilla voidaan varmistaa etteivät poikkeustilanteet, kuten yhteyskatkokset aiheuta ongelmia tiedon tulkinnassa.

Arvojen jaksollista lukemista varten asiakas voi määrittää ryhmälle kolme parametria. Luomalla tietoalkion asiakas voi päättää otetaanko niistä saadut arvot mukaan automaattiseen tiedonkeruuseen määrittämällä oliot joko aktiivisiksi tai epäaktiivisiksi.

Palvelin tarkistaa ryhmän tietoalkioiden arvot päivityssyklin (update rate) mukaisin väliajoin ja lähettää asiakkaalle muuttuneet arvot. Erottelukyky (percent deadband) on prosentuaalinen arvo ja se määrää milloin arvot on tarpeellista lähettää. Päivityssyklillä ja erottelukyvyllä on täten suuri vaikutus datan ajantasaisuuteen ja tarkkuuteen.

OPC Data Access 3.0

OPC Data Access 3.0-spesifikaatiossa tuotiin seuraavia uusia toiminnallisuuksia ja parannuksia:

- Palvelimen tunnistamisprosessiin lisätty CategoryId, josta asiakas voi päätellä mitä toiminnallisuuksia palvelin tarjoaa.
- Sovelluksissa, joissa DA-palvelinta käytetään I/O-alustana PC-valvontajärjestelmän yhteydessä, voidaan lukea ja kirjoittaa dataa luomatta OPCGroup ja OPCItem -olioita.
- Erottelukynnys ja päivityssykli voidaan asettaa OPCItem -tietoalkioon, sen lisäksi, että se jo on OPCGroup-olion asetuksissa. Tällöin ryhmän sisäisillä tietoalkioilla ei ole pakko olla samat arvot vaan asetuksista saadaan tarkemmat ja tarkoituksen mukaiset.
- Browser-rajapinta, joka helpottaa palvelimen osoiteavaruuden selausta asiakkaan toimesta.
- Yhteyden valvonta asiakkaan ja palvelimen välillä.

OPC DA 3.0:aa käyttävien tuotteiden tulee olla taaksepäin yhteensopivia ja tukea vanhempia OPC DA -versioita.

4.3 OPC Alarms and Events

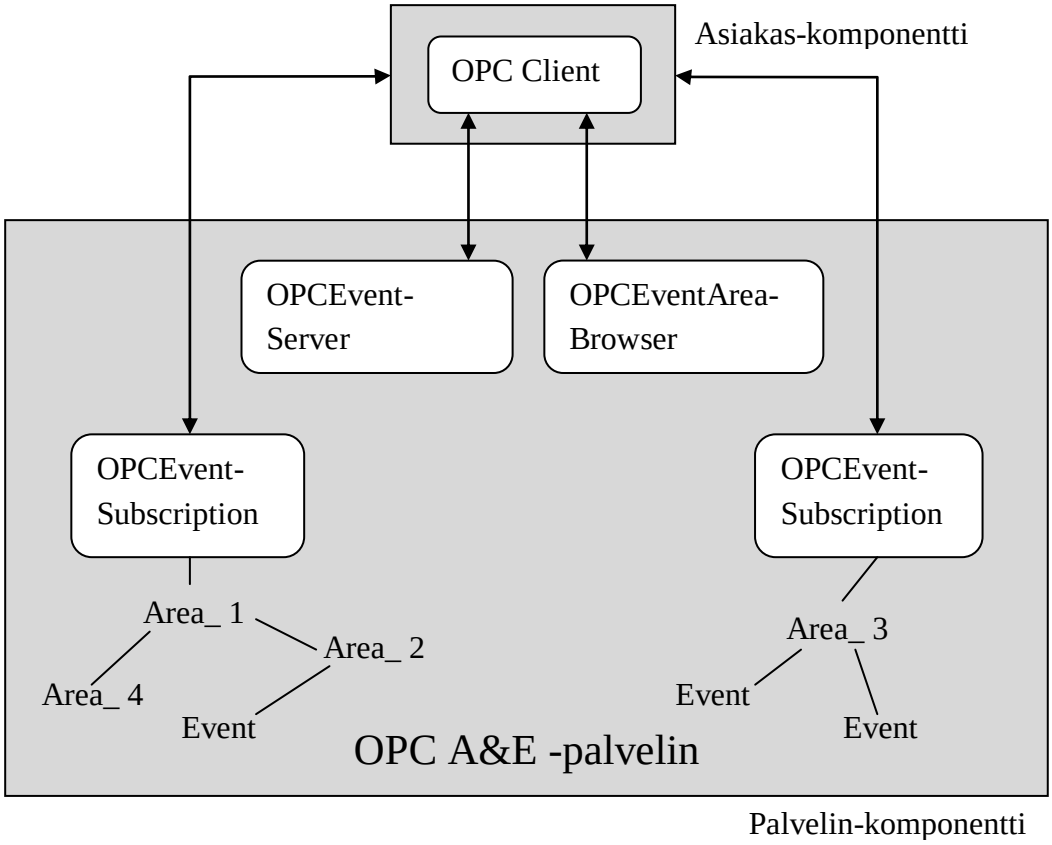
OPC Alarms and Events -rajapinnan tehtävänä on tiedottaa asiakasta prosessissa ilmenevistä normaaleista tapahtumista (Events) sekä hälytyksistä (Alarms). Normaaleja tapahtumailmoituksia on kahdenlaisia: yksinkertaiset tapahtumailmoitukset (simple event notification) ja muutostapahtumailmoitukset (tracking event notification).

Yksinkertaiset tapahtumailmoitukset käsittävät kaikki muut ilmoitukset, jotka eivät kuulu hälytysilmoituksiin tai muutostapahtumailmoituksiin. Tällaisia ilmoituksia ovat esimerkiksi laitevikailmoitukset.

Muutostapahtumailmoitukset syntyvät, kun asiakas on yhteydessä palvelimen hallinnoimiin olioihin ja aiheuttaa toiminnallaan muutoksia prosessiin, kuten vaikuttamalla automaatiojärjestelmän ohjearvoihin (setpoint).

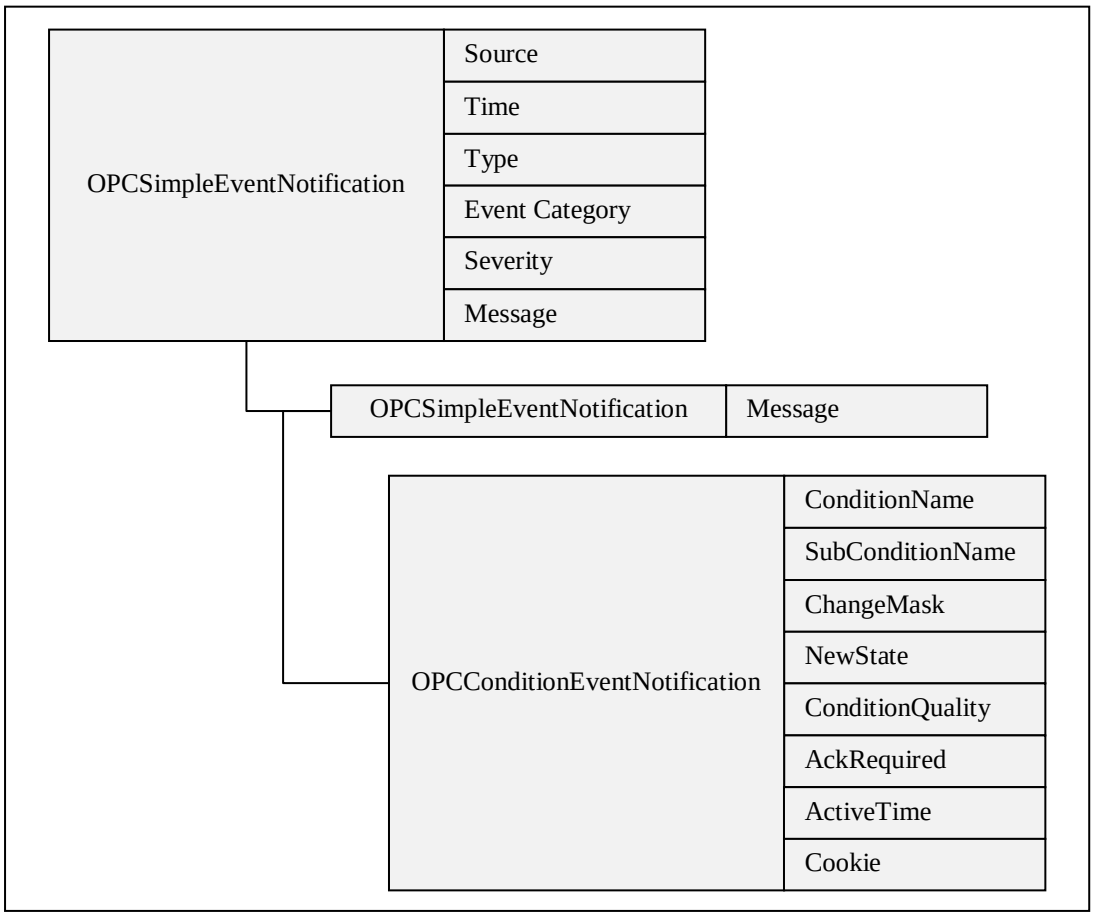
Lisäksi on myös hälytysilmoituksia (condition event notification), jotka aiheutuvat, kun jokin järjestelmän mittaama arvo ei pysy ennalta asetettujen rajojen sisällä. Tyypillisesti tällainen arvo on hyvin yksinkertainen sensorin mittaama arvo, kuten lämpötila. Hälytysilmoituksiin vaaditaan asiakkaan kuittaus, jolla varmistetaan, että hälytys on huomioitu.

OPC A&E:n tiedonsiirrossa käytetyt komponentit ja hierarkkinen rakenne ovat toiminnaltaan OPC DA:ta vastaava. Palvelimen käynnistämisen jälkeen asiakkaalla on pääsy OPCEventServer-olion rajapintoihin, jonka jälkeen asiakas voi kerätä tietoa tapahtuma-alueesta OPCEventAreaBrowserin rajapintojen käyttäen. Vaihtoehtoisesti asiakas tehdä tilauksia haluamistaan tapahtumista luomalla OPCEventSubscription-olion. Komponentit sekä hierarkkinen rakenne on esitetty kuvassa 6.



Kuva 6: OPC A&E:n komponentit ja komponenttien sisältämät oliot

OPCEventSubscription sisältää toiminnallisuuden, jolla voidaan rajata vastaanotettavia tapahtumia määrittämällä tapahtumille ehdot (filters). Ehdot koostuvat tapahtuma-alueesta (Area), sekä ehtoavaruudesta. Ehtoavaruus käsittää osan tapahtumien attribuuteista, joiden perusteella asiakas voi rajata haluamiansa tapahtumailmoituksia. Alla olevassa kuvassa 7 on esitetty kaikki tapahtumien pakolliset attribuutit. Lisäksi tapahtumat voivat sisältää laitevalmistajien omia attribuutteja. Attribuutit periytyvät ylemmistä tapahtumailmoituksista, jolloin hälytysilmoitus sisältää eniten attribuutteja.



Kuva 7: Tapahtumailmoituksien rakenne

Yksinkertainen tapahtumailmoitus sisältää seuraavat attribuutit:

- Lähde (Source): Tapahtuman lähde, jonka tiedot saadaan tapahtuma-avaruudesta.
- Aika (Time): Hetki, jolloin tapahtuma ilmeni.
- Tyyppi (Type): Tapahtuman tyyppi eli mikä kolmesta eri tapahtumailmoituksesta on kyseessä.
- Tapahtuman luokittelu (Event category): Palvelinkohtaisesti tapahtumien luokittelu erilaisiksi ryhmiksi, jotka kuvaavat tapahtumaa.
- Vakavuus (Severity): Tapahtuman vakavuus, ilmoitetaan asteikolla 1-1000. Sovelluskohtaiset vakavuusarvot tulee kääntää vastaaviksi OPC-vakavuusarvoiksi, mikäli se koetaan tarpeelliseksi.
- Viesti (Message): Tapahtumaa kuvaava vapaaehtoinen viesti.

Muutostapahtumailmoitus sisältää lisäksi tapahtuman aiheuttaneen syyn numeerisen tunnisteen (ActorId). Spesifikaatio ei kuitenkaan kuvaa tunnisteen toteuttamista tai numeeristen arvojen tarkoitusta vaan ne ovat palvelinkohtaisia.

Hälytysilmoitukset sisältävät yllämainittujen lisäksi seuraavat attribuutit:

- Tilan nimi (ConditionName): Tapahtuma-alueella olevan aktiivisen tilan nimi, joka on määritetty palvelinkohtaisesti.
- Alitilan nimi (SubConditionName): Tilaa vastaavan mahdollisen alitilan nimi.

- Maskin muutos (ChangeMask): Ilmaisee miten nykyinen tila on muuttunut.
- Uusi tila (NewState): Ilmaisee päivitetyn tilan tiedot.
- Laatu (ConditionQuality): Ilmaisee tiedon oikeellisuuden.
- Kuittaus (AckRequired): Vaaditaanko ilmoitukseen kuittaus.
- Aikaleima (ActiveTime): Aika, jolloin siirtyminen ilmoituksen kuvaamaan uuteen tilaan tapahtui. Tämä arvo ei siis ole sama kuin edellä mainittu Aika (Time).
- Arvo (Cookie): Numeerinen arvo, jota asiakas käyttää ilmoituksen kuittaukseen. Palvelin yhdistää kuittauksen tapahtumaan tämän arvon perusteella.

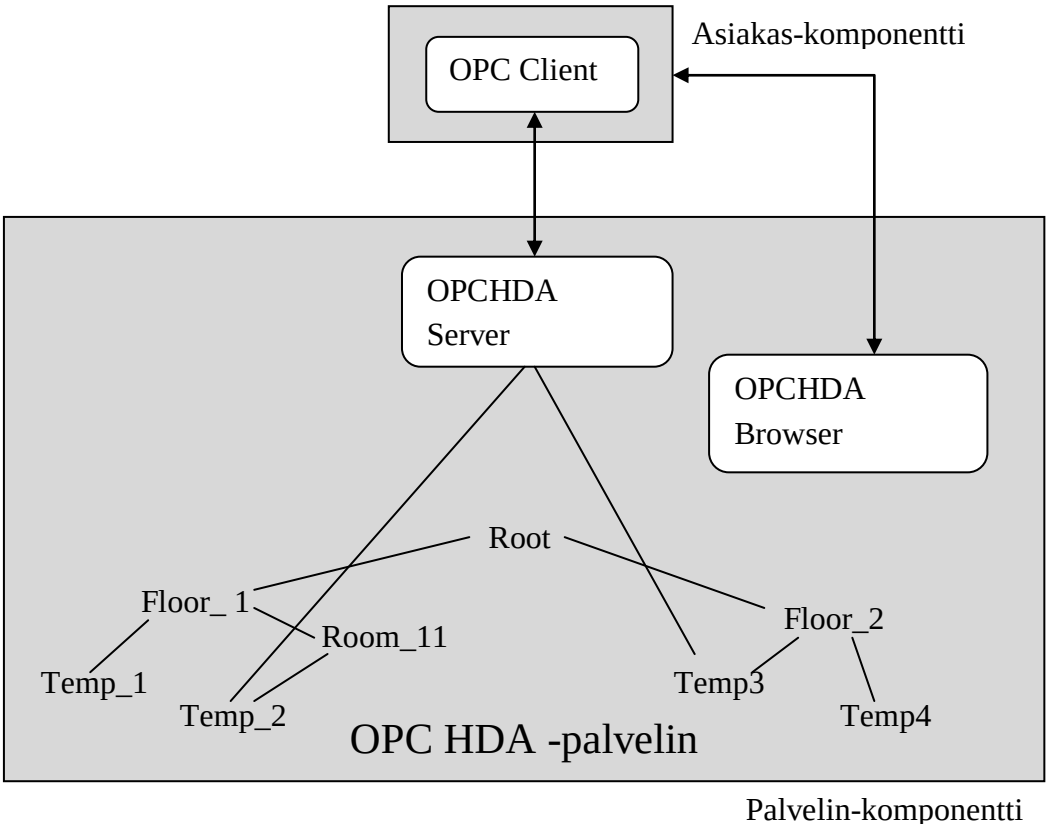
Samankaltaisuudestaan johtuen OPC A&E -rajapinta sekä OPC DA -rajapinta voidaan toteuttaa rinnakkain samassa OPC-palvelimessa, mutta tämä ei kuitenkaan ole suositeltavaa, koska muun muassa ylläpidettävyyden ja yksinkertaisuuden edut menetetään.

4.4 OPC Historical Data Access

OPC Historical Data Access (OPC HDA) -spesifikaatio määrittää rajapinnan, jonka kautta tietyllä aikavälillä kerätty data on saatavilla myöhempää käyttöä varten, eikä reaaliaikainen tiedonsiirto ole pakollista. Tällä on etunsa esimerkiksi sovelluksissa, joissa mittaavaan laitteeseen ei ole jatkuva-aikaista yhteyttä vaan data puretaan tietyin väliajoin.

OPC HDA ei kuitenkaan ota kantaa kuinka data kerätään ja tallennetaan tai miten tieto tapahtumasta saadaan. Yksi mahdollisuus on käyttää DA-asiakkaita tiedon saamiseen ja tallentaa tieto tietokantaan, jolloin se on myöhemmin HDA-palvelimen saatavilla. On huomattava, että HDA-palvelimen saatavilla voi olla valtavasti dataa, koska muuttujien määrä, sekä muuttujien sisältämien arvojen määrä, voi kasvaa prosessista riippuen todella suureksi. Yksi tärkeä HDA-palvelimen tarjoama ominaisuus on yhteen koottujen arvojen, kuten mitattujen arvojen keskiarvon tai aikavälin minimi- ja maksimiarvojen saatavuus. Nämä yhteen kootut arvot ovat prosessin historiallisen seurannan kannalta merkittävämpiä kuin muut satunnaiset, yksittäiset arvot.

Alla olevassa kuvassa 8 on esitetty OPC HDA:n komponentit. Palvelinkomponentti sisältää ainoastaan kaksi oliota; OPCHDA Server -olion, sekä OPCHDA Browser -olion. OPCHDA Browser -olion kautta asiakas voi selata HDA-palvelimen nimiavaruutta, joka sisältää kaikki datapisteet, joilla on jokin arvo. Koska dataa ei lueta kovin usein, HDA-palvelimessa ei ole erillistä oliota, jonka kautta datapisteet ovat saatavilla.



Kuva 8: OPC HDA -sovelluksen komponentit

HDA-asiakas voi lukea, tallentaa sekä muuttaa tietokannan ominaisuuksia. Tallennettuja arvoja ei voi muuttaa, mutta niitä voidaan poistaa sekä lisätä. Asiakkaalla on neljä tapaa saada historiallista dataa palvelimelta:

- Lukeminen: Asiakas voi lukea arvoja, arvojen ominaisuuksia sekä yhteen kootuja arvoja synkronisesti tai asynkronisesti.
- Päivittäminen: Asiakas voi lisätä ja korvata arvoja tietyn aikavälin sisällä tai suorittaa toiminnot tietyille arvoilla.
- Merkitseminen: Asiakas voi lukea arvon ja tehdä siihen merkinnän tai huomautuksen.
- Toisto: Asiakas voi pyytää palvelinta lähettämään tietyn aikaikkunan sisällä tallennetut arvot tietyin väliajoin. Toiminto voidaan suorittaa arvoille, sekä yhteen kootuille arvoille.

Luetut arvot sisältävät aina aikaleiman sekä tiedon datan laadusta.

5 OPC UNIFIED ARCHITECTURE

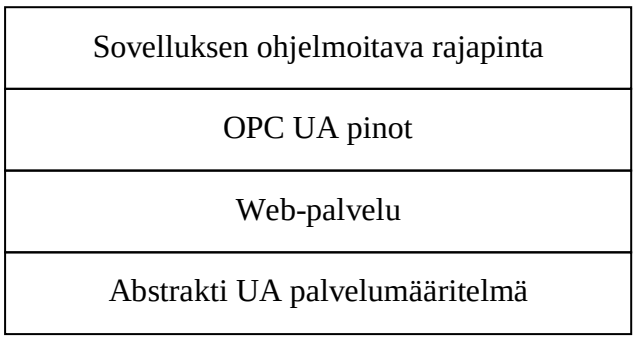
OPC UA on OPC Foundationin uusin standardi. OPC UA määrittää abstraktin joukon palveluita, joiden kautta se tarjoaa alustasta riippumattoman palvelukeskeisen arkkitehtuurin. Uuden arkkitehtuurin myötä vanhentuneesta COM/DCOM-teknologiasta on päästy eroon. Perinteisen OPC:n hyväksi todetut spesifikaatiot on OPC UA:ssa integroitu yhtenäiseksi osoiteavaruudeksi, mikä tarjoaa järjestelmätoimittajille mahdollisuuden hyödyntää laajemmin oliokeskeisiä-tekniikoita.

Tämän luvun sekä luvun 7, OPC UA:n tietoturva, lähteenä käytettiin Mahnke:n, Damn:n ja Leitner:n julkaisemaa OPC Unified Architecture ([16]) -kirjaa, joka on tehty OPC UA -spesifikaatioiden pohjalta ja sen voidaan katsoa olevan alkuperäisteos OPC UA:sta.

5.1 Palvelut

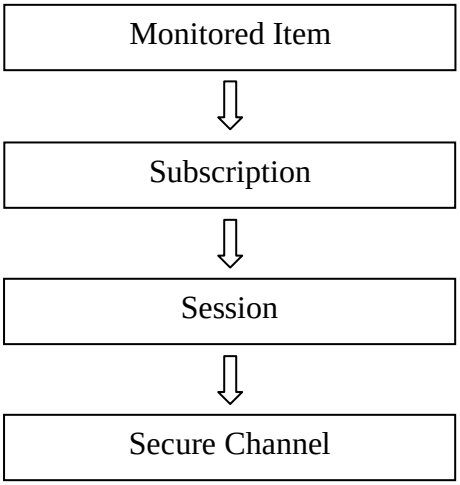
OPC UA:n palvelut määrittävät sovellustason kommunikaation asiakkaan ja palvelimen välillä. Palvelut määrittävät rajapinnat, joiden kautta asiakkaalla on pääsy palvelimen tarjoamaan dataan. OPC UA:n palvelut eivät kuitenkaan ole riippuvaisia sovellusympäristöstä, eikä käytetystä siirtoprotokollasta, mikä on oleellinen ero perinteisen OPC:n määrittämiin rajapintoihin ja Microsoft COM -riippuvuuteen nähden. Tässä kappaleessa esitellään OPC UA:n palvelut, sekä mitä etuja ne tuovat perinteiseen OPC:hen verrattuna.

Palveluiden abstrakti määritelmä mahdollistaa ohjelmointikielestä riippumattomien erilaisten siirtomekanismien toteutuksen sovellustasolla sekä palveluiden yhdistämisen verkkokerroksen kuvauksiksi. Sovelluskohtaiset ohjelmoitavat rajapinnat on määritelty OPC UA:n pinossa, joka pohjautuu UA:n palvelumääritelmään. Kommunikaatiokerrokset on kuvattu alla olevassa kuvassa 9.



Kuva 9: OPC UA kommunikaatiokerrokset

Palveluita tyypillisesti käytetään kommunikaatioon liittyvien eri tasojen luomiseen, ylläpitämiseen ja muokkaamiseen. OPC UA:n yleinen kommunikaatiorakenne on esitetty alla olevassa kuvassa 10.



Kuva 10: OPC UA kommunikaatiomallin hierarkkiset tasot

Suojattu tiedonsiirtokanava (Secure Channel) on protokollariippuvainen matalan tason tiedonsiirtokanava, jonka tehtävänä on taata turvallinen kommunikointi ja viestien vaihto. Kanava tulee luoda uudelleen tietyin väliajoin turvallisuuden takia. Uudelleenluontiväli päätetään, kun kanava luodaan ensimmäistä kertaa. Kanavan päälle luodaan istunto (Session), jonka ajaksi palvelin varaa resursseja ja vapauttaa resurssit istunnon loputtua. Istunnon yhteydessä voidaan luoda useita tilauksia (Subscription), joiden avulla asiakas ja palvelin käsittelevät dataa ja tapahtumailmoituksia. Tilauksissa voidaan luoda valvottavia kohteita (Monitored Item). Valvottavat kohteet ovat verkon solmun yksittäisiä attribuutteja, joiden muuttuvaa dataa halutaan valvoa tai ne voivat olla tapahtumailmoituksen lähteitä. Jokaiselle tasolle on määritetty palvelujoukko.

Tiedonsiirto asiakkaan ja palvelimen välillä koostuu palvelupyynnöistä ja -vastauksista. Tiedonsiirto on asynkronista, jolloin kaikki palvelut ovat määritelmän mukaan myös asynkronisia. Tällöin asiakassovelluksen ei tarvitse odottaa vastausta palvelimelta vaan voi prosessoida muita toimia sillä välin. Asynkronisuus on yksi isoista parannuksista perinteisen OPC:n synkroniseen tiedonsiirtoon nähden.

Toinen merkittävä parannus on aikakatkaisu ja virheiden käsittely, koska laajalla toiminta-alueella myös verkon toimintahäiriöt ovat todennäköisiä. Jokainen asiakkaan lähettämä palvelukutsu sisältää yksilöllisen aikakatkaisuajastimen, jonka avulla verkkovirheet on mahdollista huomata ja reagoida niihin. Virheiden käsittelyä varten on virhekoodi (StatusCode), jota käytetään virheiden huomaamiseen sekä virhekoodista lisätietoa sisältävä kuvaus (DiagnosticInformation).

Abstrakti palvelumääritelmä koostuu yleisten palvelukonseptien lisäksi seuraavista palvelujoukoista:

- Discovery Services Set - Palvelimien hakemiseen tarkoitettu palvelujoukko. Palvelujoukon tehtävänä on mahdollistaa, että asiakas löytää halutun palvelimen toimintaympäristöstä.

- Secure Channel Service Set ja Session Service Set - Palvelujoukot suojatulle tiedonsiirtokanavalle sekä istunnolle. Näiden palvelujoukkojen tehtävänä on hallinnoida eri tasojen kommunikointikanavia, jotta tiedonsiirto on turvallista, joustavaa ja luotettavaa.
- View Service Set - Palvelujoukko, jonka avulla asiakkaan on mahdollista löytää tietoa palvelimen osoiteavaruudesta. Palvelujoukko sisältää palvelut osoiteavaruudessa sijaitsevien solmujen selaamista varten, sekä metatietoa solmuista.
- Read and Write Service - Palvelua käytetään datan lukemiseen ja kirjoittamiseen. Data on solmujen attribuutteja tai muuttujien arvoja.
- Subscription Service Set ja Monitored Item Service Set - Palvelujoukkoja käytetään tapahtumien ja datan muutoksien tilaamiseen.
- Call Service - Palvelu, jota asiakas voi käyttää palvelimen menetelmien (Methods) kutsumiseen. Menetelmät ovat osoiteavaruudessa sijaitsevien objektien komponentteja, joiden avulla tiedonsiirtoa asiakkaan ja palvelimen välillä voidaan vähentää.
- HistoryRead ja HistoryUpdate Service - Palvelu, jonka avulla historiallista dataa ja historiallisia tapahtumia voidaan käsitellä.
- Query Service Set - Palvelujoukko, jota käytetään hyvin suuren osoiteavaruuden yhteydessä tiedon löytämiseen.
- Node Management Service Set - Palvelujoukko, jota käytetään palvelimen osoiteavaruuden hallinnoimiseen. Palvelujoukon avulla OPC UA -asiakkaan on mahdollista poistaa ja luoda uusia solmuja palvelimen osoiteavaruuteen.

Yllä mainitut palvelujoukot koostuvat vielä yksittäisistä palveluista, jolloin OPC UA:n palveluita on kaiken kaikkiaan 37 kappaletta, joista 21 palvelua käytetään kommunikaatiorakenteen ja yhteyksien hallinnointiin ja 16 palvelua tiedonsiirtoon.

5.2 Toimintaympäristö ja järjestelmäarkkitehtuuri

OPC UA:n toimintaympäristönä voi olla yritysverkko, tuotantoverkko tai kenttälaitteiden kontrollointiverkko. Liiketoiminta- ja tuotantojärjestelmät voivat olla Windows- tai UNIX-pohjaisia ja kontrollerit vaativat reaaliaikaisia järjestelmiä. Suunnittelun lähtökohtana onkin ollut, että OPC UA:ta voidaan hyödyntää hyvin laajassa toimintaympäristössä laitealustasta riippumatta. Sen lisäksi OPC UA:ta voidaan käyttää järjestelmätasolla arkkitehtuuristen konseptien hyödyntämisessä.

5.2.1 Palvelinmallit

Palvelinmallit on OPC UA:ssa jaettu neljään eri malliin, joissa on kuvattu tyypilliset tiedonsiirtotilanteet asiakkaan ja palvelimen kannalta. Malleja voidaan käyttää OPC UA -järjestelmien ja -sovellusten arkkitehtuurin suunnitteluun tai suunnittelun ongelmakoh-

tien ratkaisemiseen. Tässä luvussa kuvataan kyseiset palvelinmallit lyhyesti, koska mallit ovat yleisesti käytössä myös muissa sovelluksissa.

Kaikista yleisin ja yksinkertaisin malli on asiakas-palvelinmalli. Palvelin tarjoaa joukon palveluita, joita asiakas voi hyödyntää omissa toiminnoissaan pyytämällä niitä palvelimelta. Kommunikaatio asiakkaan ja palvelimen välillä muodostuu palvelupyyntö-vastaus-pareista.

Ketjutetussa palvelinmallissa on kolme osapuolta: asiakas, palvelin, sekä asiakas-palvelin-yhdistelmä. Asiakas-palvelin-yhdistelmä sisältää sekä palvelin- että asiakasrajapinnat, jolloin palvelin voi lähettää palvelupyynnön toiselle palvelimelle. Asiakas kommunikoi asiakas-palvelin-yhdistelmän kanssa ja asiakas-palvelin-yhdistelmä kommunikoi tavallisen palvelimen kanssa. Tästä on hyötyä esimerkiksi silloin, kun asiakkaalla on tarve tavallisen palvelimen tarjoamille palveluille, mutta ei pysty olemaan siihen suoraan yhteydessä. Tällöin asiakas-palvelin-yhdistelmä voi toimia yhdyskäytävänä näiden välillä.

Palvelin-palvelinmalli on lähes samanlainen kuin ketjutettu palvelinmalli, mutta siinä molempiin palvelimiin on sulautettu asiakasrajapinta. Palvelimet voivat kommunikoida keskenään ja asiakas voi kommunikoida molempien palvelimien kanssa. Palvelin-palvelinmallin yksi tyypillisimmistä käyttökohteista on palvelimien peilaus, jolloin palvelut ovat asiakkaan saatavilla, vaikka toinen palvelimista kaatuisi. Palvelimien on siis voitava kommunikoida keskenään, jotta peilaus on mahdollista ja palvelimet sisältävät saman datan.

Myös neljäs palvelinmalli, aggrekoitu palvelin, on ketjutetun palvelinmallin kaltainen, mutta yhdyskäytävänä toimiva asiakas-palvelin-yhdistelmä on yhdistetty useampaan palvelimeen, joilta saatujen palveluiden perusteella se muodostaa vastauksen asiakkaalle. Aggrekoitu palvelin siis käsittelee muilta palvelimilta saatua dataa ja valmistelee sen asiakkaalle sopivaksi. Ketjutettu palvelin vain toimittaa saadun datan eteenpäin.

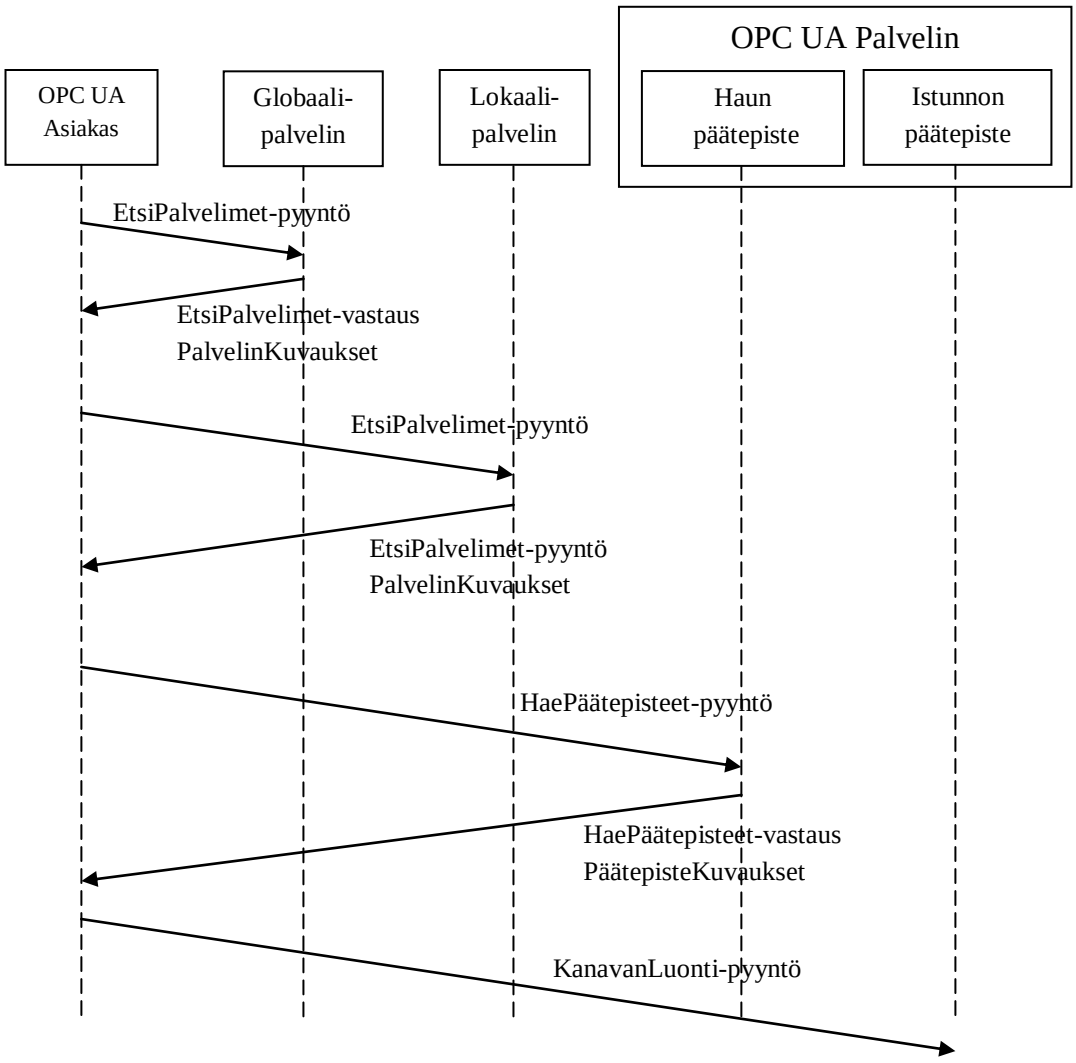
5.2.2 Päällekkäisyys / redundanssi

Luotettavuuden lisäämiseksi OPC UA:ssa on mahdollista käyttää päällekkäisiä asiakas- ja palvelinsovelluksia. Päällekkäisyydellä tarkoitetaan, että asiakas ja/tai palvelinkomponentille on olemassa duplikaatti poikkeustilanteen varalle. Tyypillisesti päällekkäisyyttä halutaan käyttää hyvin kriittisien prosessien ohjausjärjestelmissä.

Duplikaatin tehtävänä on valvoa asiakkaan ja palvelimen välistä istuntoa ja mikäli poikkeustilanteen takia asiakas- tai palvelinkomponentti vikaantuu, ohjataan istunto duplikaatille. Asiakaskomponentin vikaantuessa palvelin joutuu puskuroimaan dataa jonkin aikaa ennen kuin istunto on saatu ohjattua duplikaatille, jottei dataa häviäisi. Palvelinkomponentin vikaantuessa istunnon ohjaaminen palvelinduplikaatille voi vaatia toimia asiakkaalta, mutta vaihto voidaan suorittaa myös niin, että asiakas ei sitä huomaa. Mikäli istunnon ohjaaminen duplikaatille vaatii toimia myös asiakkaalta on OPC UA:ssa määritetty vikaantumisen varalle erilaisia toipumiskäytäntöjä.

5.2.3 Palvelinten löytäminen

Ennen kuin asiakas voi muodostaa yhteyden palvelimiin, on asiakkaan ensiksi löydettävä toimintaympäristössä olevat palvelimet, sekä niiden päätepisteet. OPC UA:n toimintaympäristössä on tyypillistä, että palvelimet on hajautettu eri verkkosegmentteihin ja niiden tiedonsiirtokäytännöt poikkeavat toisistaan. Tätä varten OPC UA:ssa on määritetty abstrakti palvelujoukko helpottamaan palvelimien löytämistä. Palvelinten löytäminen perustuu hierarkkiseen rakenteeseen, mikä muistuttaa hyvin paljon muissa tietoliikennetekniikoissa käytettyjä ratkaisuja. Alla olevassa kuvassa 11 on esitetty asiakkaan ja palvelinten välinen kommunikaatio.



Kuva 11: Asiakkaan lähettämät palvelupyynnöt UA-palvelimen selvittämiseksi, sekä palvelimien lähettämät vastaukset

Kuvan mukaisesti OPC UA -palvelimilla on kaksi päätepistettä: istunnon päätepiste (Session Endpoint) ja haun päätepiste (Discovery Endpoint). Istunnon päätepistettä käytetään suojatun tiedonsiirtokanavan (Secure Channel) ja istunnon luomiseen. Haun päätepiste tarjoaa tiedon istunnon päätepisteestä. Lisäksi on olemassa lokaaleja ja globaaleja palvelinten hakemiseen tarkoitettuja palvelimia (Local Discovery Server, Global Discovery Server). Lokaalipalvelin säilyttää tiedon palvelimista, jotka ovat samassa lait-

teessa kuin se itsekin. Globaalipalvelin puolestaan säilyttää tiedon koko verkon palvelimista. Molemmilla palvelimilla on hyvin tiedossa olevat osoitteet, jotka yleensä määritetään asiakassovellukseen. Tällöin asiakas tietää, mistä osoitteista se voi pyytää tietoa UA-palvelimista.

Yksinkertaisimmillaan asiakas saa tiedon istunnon päätepisteestä suoraan UA-palvelimelta, mikäli se tietää palvelimen osoitteen. Tällöin asiakas lähettää pyynnön palvelimen haun päätepisteelle, johon palvelin vastaa lähettämällä tiedon sopivasta istunnon päätepisteestä. Tiedon saatuaan asiakas valitsee sopivan istunnon päätepisteen ja muodostaa yhteyden palvelimeen päätepisteen kautta.

Mikäli asiakas tietää ainoastaan laitteen, jolla UA-palvelin on, mutta ei palvelimen osoitetta, joutuu asiakas pyytämään sitä lokaalilta palvelimelta. Asiakas lähettää pyynnön, johon lokaalipalvelin vastaa lähettämällä listan UA-palvelimista, jotka sijaitsevat samassa laitteessa. Lista sisältää kuvauksen palvelimista ja palvelimien osoitteet. Jos haluttu palvelin on listattuna, asiakas muodostaa siihen yhteyden edellä mainitulla tavalla.

Globaalia palvelinta tarvitaan, kun asiakkaalla ei ole tiedossa halutun UA-palvelimen osoitetta, eikä laitetta, jossa palvelin sijaitsee. Globaalipalvelin vastaa asiakkaan pyyntöön lähettämällä sille listan tiedossa olevista lokaaleista palvelimista, sekä UA-palvelimista. Mikäli haluttu palvelin löytyy listasta suoraan, asiakas lähettää pyynnön suoraan UA-palvelimelle, muuten asiakas lähettää pyynnön lokaalille palvelimelle ja menettely jatkuu aiemmin kuvatulla tavalla.

5.2.4 Jäljitettävyys / auditointi

Tapahtumien jäljitettävyys on tietoturvallisuuden kannalta hyvin merkittävä tekijä. OPC UA:n yhteydessä jäljitettävyydellä viitataan OPC UA -sovellusten normaaleihin ja epänormaaleihin toimintoihin. OPC UA:ssa on kaksi tapaa auditoida sovellusten tapahtumia: auditointilogit (Audit logs) ja auditointitapahtumien tilaaminen (Audit Events). Auditointitavat eivät ole toisiaan poissulkevia vaan molempia on mahdollista hyödyntää yhtä aikaa.

Auditointilokeihin tallennetaan OPC UA -sovellusten generoimat auditointitapahtumat. Sovellus generoi auditointitapahtuman, kun siihen kohdistuu jokin ennaltamäärätty toiminto. Tällainen toiminto voi esimerkiksi olla asiakkaan kirjautuminen palvelimelle, jolloin palvelin generoi auditointitapahtuman ja tallentaa sen lokitiedostoksi. Sekä asiakas, että palvelin voi generoida auditointitapahtumia. Auditointitapahtumia on mahdollista tilata muiden tapahtumien tapaan, mikä on hyödyllistä etenkin poikkeustilanteissa, jolloin tapahtumien tilaaja saa välittömästi tiedon tapahtumasta.

5.3 Sovellusarkkitehtuuri

OPC UA:n toiminnallisuuksien kannalta toimittajien tulee ottaa huomioon omissa OPC UA -sovelluksissa tietyt suunnitteluperiaatteet tietoturvallisuuden, suorituskyvyn, siir-

rettävyyden sekä uudelleenkäytettävyyden kannalta. Tässä luvussa kuvataan OPC UA sovellusarkkitehtuurin periaatteet abstraktilla tasolla.

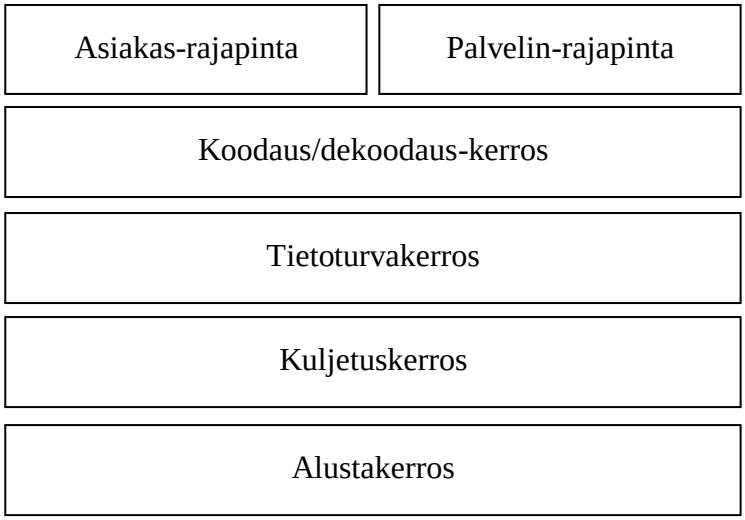
Sovellusarkkitehtuurin periaatteena on, että sovellus koostuu toiminnallisista kerroksista, joilla on tietyt ominaiset tehtävät. Kerrokset on nimetty pinoksi, SDK-kerrokseksi (Software Development Kit) ja sovelluskerrokseksi ja ne on esitetty alla olevassa kuvassa 12.

Kuva 12: Sovellusarkkitehtuurin toiminnalliset kerrokset

Kuvan mukaisesti sovelluksen toiminnallisuus on jaettu tapauskohtaisiin toimintoihin, sekä perustoiminnallisuuksiin, jotka koostuvat matalan tason ja korkean tason toiminnoista. Lisäksi OPC Foundation tarjoaa niin sanottua UA SDK-pakkausta, joka sisältää sovelluskehittäjille valmiita pinoja, kirjastoja ja esimerkkisovelluksia, mutta niitä ei tässä työssä tarkemmin käsitellä.

5.3.1 Pino

Pino on OPC UA -sovellusten yhteinen osa, joka kattaa matalan tason toiminnallisuuden. Pino koostuu asiakas- ja palvelin rajapinnoista sekä neljästä eri kerroksesta, joista jokaisella on oma tehtävänsä: koodaus/dekoodaus-kerros (encoding), tietoturvakerros (security), kuljetuskerros (transport) ja alustakerros (platform). Kerrokset ovat esitetty kuvassa 13.



Kuva 13: Pinon sisältämät kerrokset

Sovellus- ja SDK-kerrokset ovat yhteydessä pinoon asiakas- ja palvelin rajapintojen kautta, jotta ne voivat lähettää ja vastaanottaa viestejä. Rajapinnat käyttävät samaa pinoa, koska pino tarjoaa paljon perustoiminnallisuuksia, mutta molemmille rajapinnoille on myös ominaisia toiminnallisuuksia, minkä takia rajapinnat on toteutettu erikseen.

Rajapinnalta viestit tulevat koodaus/dekoodaus-kerrokselle, jossa viestien koodaus ja dekoodaus suoritetaan. Vastaanotettu viesti muunnetaan tavujonoksi (serialized) OPC UA:n sääntöjen mukaisesti, jonka jälkeen se annetaan tietoturvakerrokselle. Vastaavasti tietoturvakerrokselta saatu tavujono muunnetaan alkuperäiseen muotoon ja annetaan viesti ylemmälle kerrokselle.

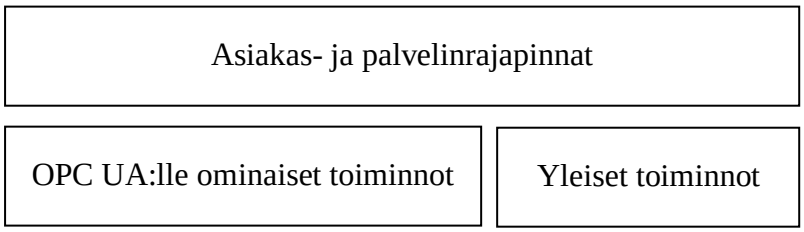
Tietoturvakerros suojaa koodaus/dekoodaus-kerrokselta saadut koodatut viestit joko allekirjoituksella tai allekirjoituksella sekä salauksella. Suojauksesta riippuen vastaanottaja ensiksi purkaa salauksen ja sen jälkeen verifioi allekirjoituksen viestiin liitettyjen tietoturvaotsakkeiden ja -alaviitteiden perusteella. Täysin eristetyissä ympäristöissä on myös mahdollista jättää viestit täysin suojaamatta.

Kuljetuskerros vastaa viestien lähettämisestä ja vastaanottamisesta. Erilliset siirto-otsakkeet liitetään viestiin ennen viestin lähettämistä, ja ne sisältävät erityistä tietoa viestistä, kuten viestin tyypin ja pituuden. Vastaanottajan siirtokerros tarkistaa, että viesti on tunnistettavissa sekä ettei viesti ole liian pitkä.

Alustakerros sisältää kaiken alustalle ominaisen koodin. Tällöin pinon muut osat voidaan uudelleen käyttää alustaa vaihdettaessa ja ainoastaan alustakerros täytyy vaihtaa.

5.3.2 SDK-kerros

SDK-kerros kattaa korkeamman tason perustoiminnot ja se koostuu kolmesta komponentista alla olevan kuvan 14 mukaisesti.



Kuva 14: SDK-kerroksen komponentit

OPC UA:lle ominaisiin toimintoihin kuuluu OPC UA:n määrittämät konseptit ja palvelut, kuten istunnot (Sessions), tilaukset (Subsciptions) ja tapahtumat (Events).

Yhteiset ominaisuudet pitää niin ikään toteuttaa sekä asiakkaan että palvelimen päädyssä. Yhteisiin ominaisuuksiin kuuluvat sovellusten konfiguraatiot ja kirjaukset, sekä tapauskohtaisesti myös taata mahdollisuus sovellusten käyttämien sertifikaattien oikeellisuuden tarkistamiseen.

Sertifikaattien oikeellisuuden tarkistaminen on mahdollista toteuttaa myös Pinossa, jos tarkistamismenettelyt ovat samanlaiset kaikilla sovelluksilla. Validointi on

kuitenkin järkevämpää suorittaa SDK-kerroksella tai jopa sovelluskerroksella, mikäli menettelytavat vaihtelevat toimintaympäristössä.

Asiakas- ja palvelinrajapinnat mahdollistavat tiedonsiirron sovelluskerroksen ja SDK-kerroksen välillä. Asiakasrajapinnat vastaavat palvelupyyntöjen lähettämisestä ja vastauksien vastaanottamisesta. Palvelinrajapinnat puolestaan SDK-kerroksen alustamisesta ja konfiguroinnista, sekä dataa tarjoavien taustajärjestelmien integroinnista.

5.3.3 Sovelluskerros

Sovelluskerroksen arkkitehtuuri voi tapauskohtaisesti vaihdella todella paljon, mutta periaatteessa sovellukset ovat joko asiakassovelluksia tai palvelinsovelluksia. Asiakassovellusten tehtävänä on usein tarkastella palvelimen osoiteavaruutta ja visualisoida SKD-kerroksen tarjoamaa dataa, sekä kääntää käyttäjän tekemiä toimintoja SDK:n rajapinnoille sopiviksi pyynnöiksi.

Palvelinsovelluksia on kahdenlaisia. Toinen näistä hallinnoi palvelimen päämuistissa olevaa osoiteavaruutta ja lataa palvelimen käynnistyessä osoiteavaruuden päämuistiin. Osoiteavaruus voi olla tallennettuna esimerkiksi tietokantaan tai yksittäiseen tiedostoon.

Toisen palvelinsovelluksen tehtävänä on hoitaa pääsy taustajärjestelmiin, jotta tieto taustajärjestelmien osoiteavaruuksista voidaan kerätä. Taustajärjestelmät ovat tyypillisesti laitteita tai kontrollereita.

5.4 Profiilit

OPC UA:n toiminnallisuuksien joukko on laaja ja kaikki OPC UA -sovellukset eivät sisällä kaikkia toiminnallisuuksia. OPC UA sisältää neljään eri kategoriaan liittyvää profiilia, jotka määrittävät sovelluksen toiminnallisuuden: asiakassovelluksen profiilit, palvelinsovelluksen profiilit, siirtoprofiilit ja tietoturvaprofiilit. Profiilien avulla voidaan testata ja todentaa sovelluksen toiminnalliset ominaisuudet ja tätä varten on olemassa itsenäisiä viranomaistahoja, jotka myöntävät sovelluksille sertifiointeja.

5.4.1 Asiakas- ja palvelinsovelluksien profiilit

Palvelinsovelluksen profiileja on kahdenlaisia: täydet ominaisuudet käsittäviä profiileja (full-featured) ja yksittäisiä lisäprofiileja (facet). Täyden ominaisuuden profiilit ovat sellaisia, joita odotettavasti suurin osa sovelluksista tukee ja palvelimen tulee tukea ainakin yhtä täydet ominaisuudet käsittävää profiilia. Lisäprofiilit tuovat yksittäisiä ominaisuuksia ja niitä käytetään täyden ominaisuuden profiilien rinnalla täydentämään palvelimen toiminnallisuuksia.

Asiakassovelluksen profiilit ovat aina yksittäisiä lisäprofiileja, koska odotettavasti asiakassovellukset harvoin tukevat samoja profiiliryhmiä, minkä takia täyden ominaisuuden profiileille ei ole tarvetta.

5.4.2 Kuljetusprofiilit

Kuljetusprofiilit ovat kaikki yksittäisiä profiileja ja ne määrittävät sovelluksen tukemat tiedonsiirtoprotokollat: kuljetuskerroksen protokollat SOAP/HTTP ja UA TCP, tietoturvakerroksen protokollat WS-SecureConversion ja UA-SecureConversion, sekä koodaus/dekoodaus-kerroksen protokollat UA XML ja UA Binary. Profiileja on viisi kappaletta, joista jokainen on yksi näistä protokollista muodostettu järkevä yhdistelmä.

Palvelinsovellukset luonnollisesti tukevat mahdollisimman suurta kuljetusprofiilien joukkoa, jotta ne voivat palvella hyvin monen eri toimialueen asiakkaita. Asiakassovelluksien tarvitsee tukea ainoastaan omalla toimialueella käytettäviä profiileja.

5.4.3 Tietoturvaprofiilit

Tietoturvaprofiilit määrittävät algoritmit ja salausavainten pituudet sertifiointien oikeellisuuden tarkistamiseen, viestien salaamiseen ja allekirjoittamiseen. Tällä hetkellä profiileja on kolme kappaletta: Basic128Rsa15, Basic256 ja ilman salausta (None), mutta odotettavasti ne tulevat muuttumaan tulevaisuudessa joko pidentämällä salausavainta tai muuttamalla algoritmeja monimutkaisemmiksi, jotta salausta ei ole mahdollista murtaa raa'alla laskentateholla.

5.5 OPC:n konversio OPC UA:ksi

Perinteinen OPC on ollut hyvin menestyksekäs ja se on edelleen laajasti käytössä. Tämän takia OPC UA on suunniteltu taaksepäin yhteensopivaksi, jotta se pystyy hyödyntämään perinteisen OPC:n hyväksi osoittautuneita konsepteja sekä järjestelmämuutoksen helpottamiseksi. Perinteinen OPC on mahdollista konvertoida OPC UA:ksi menettämättä tietoa muunnoksessa. Konversion päämääränä on, että OPC-asiakas voi kommunikoida UA-palvelimen kanssa ja UA-asiakas voi kommunikoida OPC-palvelimen kanssa. Konversio mahdollistaa myös OPC-tuotteiden yhdistämisen OPC UA:iin, jolloin UA:n tietoturvakäytäntöjä voidaan hyödyntää ilman uusien ominaisuuksien toteuttamista OPC-tuotteisiin. Konversio on mahdollista tehdä myös toisinpäin, mutta se ei ole suositeltavaa, koska tietoa voidaan menettää muutoksessa.

Tässä luvussa käydään perinteisen OPC:n konversio OPC UA:ksi pääpiirteittäin läpi eri spesifikaatioiden tapauksissa. Kaikki muuttujat, niiden ominaisuudet ja attribuutit, sekä konversiot löytyvät tarkemmin listattuna OPC Foundationin OPC UA Specification 1.02, Part 6: Mappings -osiosta.

5.5.1 OPC Data Access

Kenties tärkein konvertoitava spesifikaatio on OPC DA. Konversio ei ole monimutkainen vaan lähtökohtana on, että OPC DA:n osoiteavaruus ja tiedon käsittely saadaan kuvattua OPC UA:n omilla komponenteilla.

OPC DA:n osoiteavaruuden kuvaamiseen käytetään OPC UA:n pääkomponentteja, jotka ovat Folder Object, Data Variables, Organizes References ja HasComponent References. Folder Objectit kuvaavat OPC DA:n hierarkkisen osoiteavaruuden haaroja ja toimivat hierarkian juurena. Organizes Refences järjestävät osoiteavaruuden hierarkian. Data Variablesit esittävät OPCItem-olioita ja niiden muodostamiseen käytetään Folder Object ja HasComponent References-komponentteja.

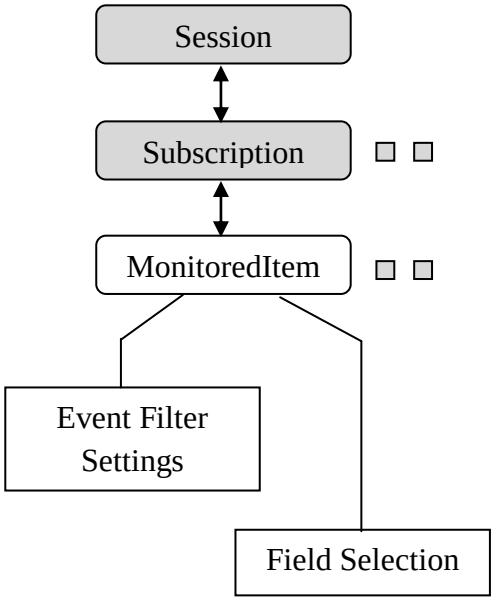
Vastaavasti yhteyden muodostamista ja tiedon käsittelyä varten on UA:ssa OPC DA:n vastineet. OPCServer-palvelinoliota kuvaa OPC UA Session, OPCGroup-oliota OPC UA Subscription ja OPCItem-oliota Monitored Item. Vastinparit on esitetty kuvassa, joka on luvussa 5.5.4.

5.5.2 OPC Alarm & Events

OPC A&E:n malli on paljon DA:ta rajoitetumpi ja staattisempi, minkä takia sen kuvaaminen OPC UA:ksi ei ole aivan yhtä suoraviivaista, vaikka periaate on sama. Yksinkertaiset ja muutostapahtumailmoitukset voidaan toteuttaa sellaisinaan OPC UA:ssa, koska niiden tilaaminen ja valvonta kuuluvat OPC UA:n perustoimintoihin.

OPC UA:ssa on Alarms & Conditions -tietomalli, jota tarvitaan hälytysilmoitusten tilaamiseen ja valvontaan. Alarms & Conditions -malli on kuitenkin tällä hetkellä vasta suunnitteluvaiheessa. Ideana on kuitenkin, että tapahtumailmoitusten lisäksi otetaan käyttöön tilalliset ilmoitukset (Conditions), jotka paremmin sopivat hälytysilmoituksiksi, koska ne säilyttävät tilan, eivätkä ole ohimeneviä, kuten muut tapahtumailmoitukset.

Osoiteavaruus kootaan hierarkkisesti ja samalla määritetään tapahtumatyypit, sekä tilalliset ilmoitukset. Alla olevassa kuvassa 15 on esitetty rakenne, jonka mukaan tapahtumia valvotaan ja tilataan.



Kuva 15: OPC UA:n tapahtumien tilaus ja valvonta

Siinä missä OPC A&E:n OPCEventSubscription-olio sisälsi ainoastaan yhden suodattimen tapahtumailmoituksille, UA:n Subscription-olio voi sisältää useampia MonitoredItem-olioita, joista jokainen sisältää asetukset haluttujen tapahtumien suodattamiseen. Lisäksi UA:ssa ei ole ennalta määritettyjä tapahtumatyypille ominaisia attribuutteja, vaan asiakas voi valita halutut kentät kaikista tapahtumakentistä.

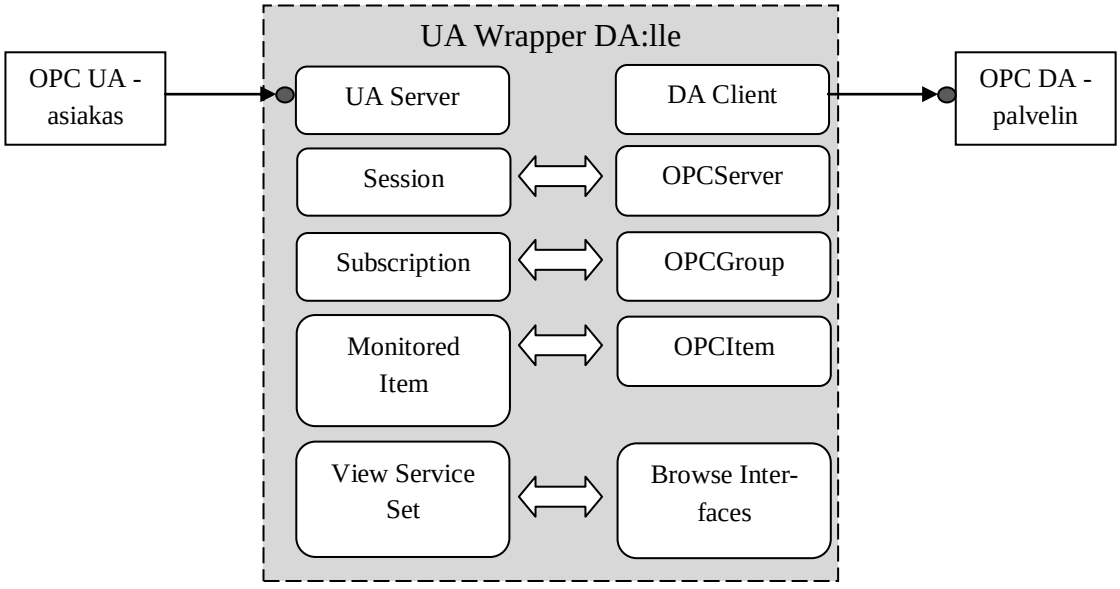
5.5.3 OPC HDA

OPC HDA:n komponentit, osoiteavaruus ja tiedon käsittely on hyvin suoraviivaisesti kuvattu OPC UA:ssa. Osoiteavaruuden kuvaamisessa tarvitaan hyvin pientä osaa UA:n mahdollisesti kapasiteetista ja tiedon käsittelyn kannalta muutokset oliokuvauksissa ovat lähinnä nimellisiä. Ainoa toiminnallinen eroavaisuus on UA:n tukema tapahtumahistoria (Event History).

5.5.4 Wrapperit ja proxyt

Wrapperit ja proxyt mahdollistavat perinteisen OPC:n ja OPC UA:n asiakkaiden ja palvelinten välisen kommunikaation ilman muutoksia alkuperäisiin OPC-tuotteisiin. Perinteisen OPC:n rajapinnat muunnetaan OPC UA:n rajapinnoiksi ja päinvastoin.

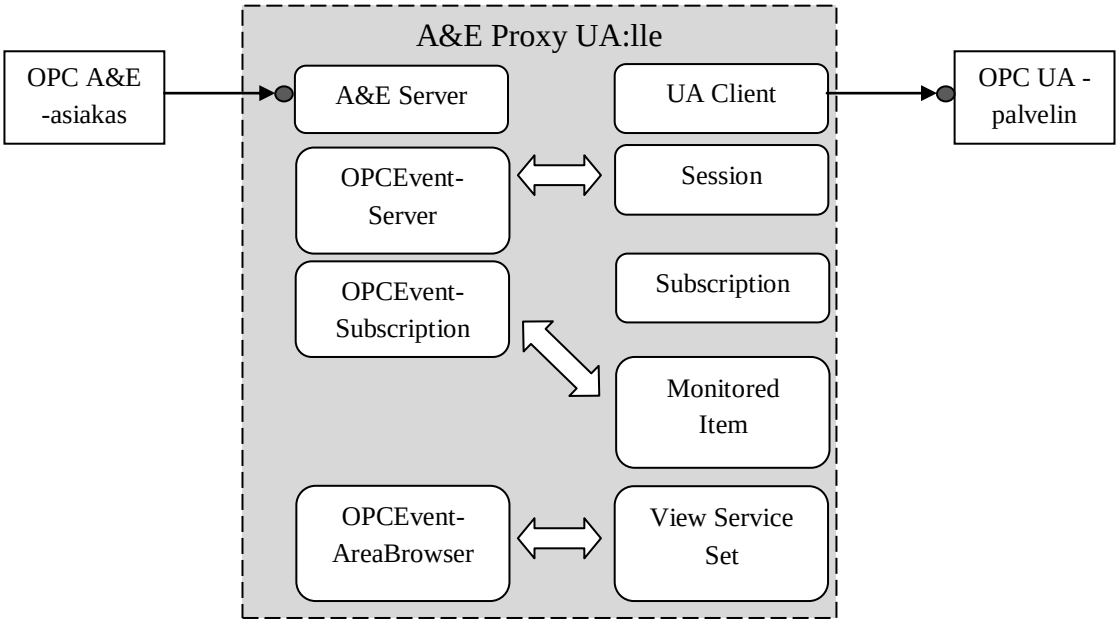
Wrappereita käytetään, jotta OPC UA -asiakas voi kommunikoida OPC-palvelimen kanssa. Wrapper sisältää sekä OPC:n, että OPC UA:n komponentteja ja konversio näiden välillä tehdään wrapperin sisällä. Kuvassa 16 on esimerkin omaisesti esitetty UA-asiakkaan ja DA-palvelimen kommunikaatio wrapperin välityksellä.



Kuva 16: UA-wrapperin sisältämät komponentit sekä konversio, jotta UA-asiakas voi kommunikoida DA-palvelimen kanssa.

Wrapperit vaativat kuitenkin ylimääräisen ohjelmistokerroksen, jolloin niiden käyttö ei ole yhtä tehokasta kuin aiemmassa luvussa mainittu suora konversio. Wrapperien myötä myös ylläpito on vaivalloisempaa ja samalla ne aiheuttavat rajoituksia.

Proxyt toimivat wrapperien tavoin, mutta mahdollistavat tiedonsiirron OPC-asiakkaan ja OPC UA -palvelimen välillä. Kuvassa 17 on esimerkkinä esitetty OPC A&E proxy UA:lle. Myös proxyt aiheuttavat tehohäviötä, minkä takia niitä ja wrappereita ei tulisi käyttää enää uusissa järjestelmissä vaan ainoastaan vanhojen, olemassa olevien OPC-järjestelmien ja UA:n integroinnissa.



Kuva 17: OPC A&E:n ja UA:n komponentit, sekä näiden välinen konversio proxyn sisällä.

6 TIETOTURVA

Tietoturvallisuuden rooli on entistä merkittävämpi, kun teollisuusautomaatioverkot yhdistetään toimistoverkkoon. Aikaisemmin hyvin eristyksissä olleita automaatioverkkoja rakentaessa ei välttämättä olla mietitty verkkojen tietoturvapuolta ollenkaan, koska tietoturvaa ei pidetty tärkeänä tekijänä ennen 2000-lukua. Tästä johtuen hyvin suuri osa automaatioverkoista on haavoittuvaisia uusilla tietoturvauhille. [17]

Teollisuusautomaatiojärjestelmillä on joitakin yhteisiä piirteitä muiden tietojärjestelmien kanssa, jolloin samoja toimintatapoja voidaan tehokkaasti hyödyntää tietoturvaongelmiin sekä automaatio-, että toimistoympäristössä. Automaatiojärjestelmillä on myös erityispiirteitä, jotka asettavat omat haasteensa ja ne tulee huomioida tietoturvamallia suunnitellessa. Tärkeimmät erityispiirteet ja haasteet ovat [18]:

- **Vakavat seuraukset.** Toimintahäiriöt tuotantoympäristössä voivat aiheuttaa vakavia taloudellisia ja fyysisiä vahinkoja. Vakavien seurauksien takia turvallisuus on kriittisin toiminnallinen vaatimus automaatiojärjestelmissä, eikä luottamuksellisuus- ja yksityisyysongelmat, kuten muissa tietojärjestelmissä.
- **Prosessien jatkuva-aikaisuus.** Monia teollisuusautomaatiojärjestelmiä ei voida pysäyttää kuin harvoin tuotannon jatkuvan toiminnan vuoksi. Tällöin järjestelmä- ja tietoturvapäivitysten tekeminen pitkittyy, minkä takia järjestelmän käytettävyys ja robustisuus ovat merkittäviä.
- **Järjestelmien resurssit ja reaaliaikaisuus.** Toimistojen tietojärjestelmiin verrattuna automaatiojärjestelmät ovat resursseiltaan rajoittuneempia, hajautuneempia ja vaativat reaaliaikaisuutta, minkä takia tietoturvaratkaisuiden tulee olla suoritusvaatimuksiltaan kevyitä.
- **Automaatiojärjestelmien elinkaari.** Käytössä olevia automaatiojärjestelmiä voi olla monilta eri vuosikymmeniltä, jolloin ne voivat olla rajoittava tekijä tietoturvallisuuden suunnittelussa. Pitkän elinkaaren etuna on, että järjestelmät ovat pidemmälläkin aikavälillä tarkasteltuna suhteellisen heterogeenisiä. Tiedonsiirtoon ja tietoturvallisuuteen liittyvät toiminnallisuudet tulisikin uusiin automaatiojärjestelmiin suunnitella siten, että ne toimivat vielä lähitulevaisuudessa uusien protokollien ja järjestelmien kanssa.
- **Moninaiset käyttäjäryhmät.** Useat eri käyttäjäryhmät voivat käyttää tuotannon automaatiojärjestelmiä. Käyttäjäryhmien taustojen, asenteiden ja toimintamallien takia automaatiojärjestelmien hallinta- ja vastuualueet voivat olla sekavia, mikä vaikeuttaa tietoturvalliseen ratkaisuun pääsemistä.

6.1 Tietoturvatavoitteet

Vaatimukset tietoturvallisuudelle pohjautuvat aina olemassa oleviin uhkiin sekä riskianalyysiin ja suunniteltuihin vastatoimenpiteisiin. Tässä luvussa ei käsitellä riskianalyysia vaan tarkastellaan kahdeksaa tietoturvatavoitetta, jotka pätevät niin toimisto- kuin automaatioverkossakin.

6.1.1 Luottamuksellisuus

Luottamuksellisuustavoite viittaa siihen, että tietoa ei paljastu luvattomille henkilöille tai järjestelmille. Tieto voi liittyä joko tietoturvamekanismeihin itsessään, kuten salasanoihin tai esimerkiksi automaatiojärjestelmien suorituskykyyn tai tuotannon ajotietoihin.

6.1.2 Eheys

Eheydellä tarkoitetaan tiedon koskemattomuutta luvattomien henkilöiden tai järjestelmien toimesta. Tiedon koskemattomuus käsittää myös tiedonsiirrossa käytettyjen viestien eheyden eli viestejä ei voi muokata, uudelleen lähettää tai viivästyttää.

6.1.3 Saatavuus

Saatavuustavoitteen tehtävänä on varmistaa, että luvattomat henkilöt tai järjestelmät eivät voi estää valtuutettua henkilöä käyttämästä järjestelmää tai muuttaa henkilön käyttöoikeuksia.

6.1.4 Valtuuttaminen

Valtuuttamisella tarkoitetaan käyttöoikeuksien hallintaa. Sen tehtävänä on varmistaa, ettei henkilöillä tai järjestelmillä ole pääsyä tai mahdollisuutta käyttää järjestelmiä tai järjestelmien osa-alueita ilman käyttöoikeuksia. Kyseessä on siis mekanismi, joka erottaa laillisen ja luvattoman käyttäjän toisistaan.

6.1.5 Todentaminen

Todentamisen tehtävänä on tunnistaa, kuka henkilö järjestelmää yrittää käyttää ja onko henkilö oikeasti väittämänsä kyseinen henkilö. Henkilön todentamiseen käytetään käyttäjille määritettyjä tunnistetietoyhdistelmiä, kuten käyttäjätunnusta ja salasanaa, ja verrataan niitä järjestelmän tuntemiin tunnistetietoyhdistelmiin.

6.1.6 Kiistämättömyys

Kiistämättömyyden päämääränä on osoittaa ja todistaa ulkoiselle taholle, kuka on vastuussa järjestelmään suoritetuista toimista. Tavoite on tyypillisesti relevantti lainsäädännöllisten vaatimusten, kuten laatu- ja turvallisuusvaatimusten ja korvauksien yhteydes-

sä, kun tarkastellaan onko vaatimuksia noudatettu. Kiistämättömyystavoitteen laiminlyönnillä ei siis yleensä ole tietoturvallisuuteen liittyviä seuraamuksia.

6.1.7 Jäljentäminen

Jäljentämisen (auditability) tarkoituksena on, että kaikki järjestelmää koskeva oleellinen, historiallinen tieto, kuten suoritetut käskyt, tallennetaan lokeihin ja järjestelmän toimintaa voidaan tarkastella jälkeenpäin lokeista. Jäljentäminen on hyvin oleellinen osa, kun selvitetään syitä järjestelmän toimintahäiriöille ja/tai seurauksille, mikäli tietoturvaan liittyvää haavoittuvuutta on mahdollisesti jo hyödynnetty.

6.1.8 Vahingon eristäminen (third-party protection)

Tavoite viittaa siihen, että haavoittunut järjestelmä ei saa aiheuttaa vahinkoa muille järjestelmille, käyttäjille tai tiedoille, eikä etenkään kolmansille osapuolille. Varsin tyypillinen vahingon aiheuttaja on saastunut järjestelmä, joka käyttää palvelunestohyökkäystä estääkseen muiden järjestelmien toiminnan tai lähettää haittaohjelmia saastuttaakseen muutkin järjestelmät.

6.2 Tärkeimmät tietoturvavaatimukset automaatiojärjestelmille

Organisaatiot vastaavat omasta tietoturvastaan, jolloin myös automaatiojärjestelmien tietoturvavaatimukset määräytyvät organisaation tietoturvapolitiikan mukaan. Tietoturvavaatimusten yleisenä ongelmana on arvioida ja valita parhaat menetelmät, jotta tietoturvariskit saadaan minimoitua ilman, että se vaikuttaa järjestelmien käytettävyyteen. Tässä luvussa käsitellään automaatiojärjestelmiä koskevia tietoturvan perusperiaatteita.

6.2.1 Verkon syvyyssuuntainen suojaus

Verkon tietoturvaa suunnitellessa on olemassa kaksi lähestymistapaa: yksi läpäisemätön suojamuuri (Hard perimeter) tai syvyyssuuntainen suojaus (Defense in Depth). Ainoastaan jälkimmäinen on järkevä ja turvallinen valinta, sillä Hard perimeter -mallissa on monta puutetta ja ongelmakohtaa.

Hard perimeter -mallissa suojamuurin tulisi olla läpäisemätön, mikä on käytännössä mahdotonta toteuttaa, koska suojamuuri perustuu ainoastaan yhteen mekanismiin tai laitteeseen, joka voi pettää. Pettävä kohta voi löytyä suojamekanismin suunnittelusta, toteutuksesta tai toiminnasta. Mikäli suojamuuri pettää, on koko verkko täysin avoin hyökkäykselle, koska hyökkäykseen ei millään tavalla pystytä eikä ehditä reagoimaan, sillä varalla ei ole muita suojausmekanismeja. Hyökkäys ei myöskään välttämättä tule aina verkon ulkopuolelta vaan on mahdollista, että hyökkäys tapahtuu verkon sisäpuolelta. Tällöin suojamuuri on täysin hyödytön.

Defense in Depth -mallissa verkko on jaettu useisiin verkkosegmentteihin ja vyöhykkeisiin. Uloimmilla vyöhykkeillä on vähemmän tärkeät, suojattavat kohteet ja

sisemmillä vyöhykkeillä tärkeät kohteet. Tällöin uloimmat vyöhykkeet suojaavat myös sisempiä vyöhykkeitä, tässä tapauksessa automaatiojärjestelmiä, jotka ovat erittäin turvallisuuskriittisiä. Vyöhyke voidaan jakaa vielä useampaan verkkosegmenttiin, jolloin esimerkiksi eri järjestelmätoimittajien laitteet voidaan eristää omiin verkkosegmentteihin. Segmenttien ja vyöhykkeiden sisällä käytetään tyypillisesti toisistaan poikkeavia tietoturvallisuusmekanismeja, jolloin hyökkäyksen tekeminen on monimutkaisempaa ja vie enemmän aikaa. Suojamekanismien lisäksi käytetään mekanismeja, joiden avulla hyökkäykset on mahdollista tunnistaa ja niitä vastaan voidaan aktiivisesti puolustautua. [19]

6.2.2 DMZ:n käyttö

Demilitarized Zone (DMZ) on aliverkkotyyppi, jota käytetään erottamaan aliverkkoja, joiden tietoturvatasot poikkeavat toisistaan. Lähtökohtaisesti on usein parasta määritellä, että aliverkkojen välillä mitään tietoa ei saa siirtyä suoraan. Etenkin automaatioverkkoon päin tuleva suora liikenne tulee kieltää ja ulospäin lähtevä liikenne rajoitetaan vain välttämättömään tietoon. Kaikki tietoliikenne automaatioverkon ja toimistoverkon välillä tulee välittää DMZ-alueella sijaitsevien palvelinten kautta. Palvelimilla tieto tallennetaan ja esikäsitellään. Palvelimiin on pääsy ainoastaan ennalta määritetyistä tietokoneista ja sovelluksista salatulla ja käyttäjätunnistetulla yhteydellä, jolloin automaatioverkkoon kohdistuva verkon ulkopuolinen hyökkäys voidaan estää.

Toimistoverkossa havaitun uhan yhteydessä DMZ-alue voidaan irtikytkeä, jolloin automaatioverkon toiminta jatkuu normaalina ja uhka on eristetty. Vastaavasti muiden aliverkkojen kohdalla tulisi suunnitella irtikytkentätoimenpiteitä hyökkäysten eristämiseksi. [18]

6.2.3 Palomuurien käyttö

Palomuuri koostuu rautapohjaisesta laitteistosta sekä ohjelmistosta ja se auttaa tietoverkkojen pääsynhallintatehtävissä. Palomuurin käytöllä on kolme tavoitetta, joista ensimmäinen on, että kaikki liikenne verkkojen välillä ohjataan palomuurin kautta, jolloin palomuurilla on mahdollista suodattaa ja rajoittaa tietoliikennettä verkkojen välillä [20]. Palomuuria tyypillisesti käytetään erottamaan organisaation turvalliseksi luokiteltu lähiverkko turvattomaksi luokitellusta ulkoverkosta, Internetistä. Automaatio- ja toimistoverkkojen välillä tulee kuitenkin käyttää palomuuria, koska suuresta toimijamäärästä ja poikkeavasta tietoturvatasosta johtuen myös toimistoverkko luokitellaan tässä tapauksessa turvattomaksi [17]. Toisena tavoitteena on rajata verkkojen välinen liikenne vain valtuutettuun liikenteeseen. Palomuurisääntöihin määritetään organisaation tietoturvapolitiikan mukaiset sallitut yhteydet ja lähtökohtaisesti kaikki muu liikenne kielletään.
Kolmas tavoite koskee palomuuria. Palomuurin tulee olla läpäisemätön ja riittävän kestävä, mikäli se joutuu hyökkäyksen kohteeksi. Palomuuri antaa ainoastaan väärän turvallisuuden tunteen, mikäli se pettää hyökkäyksen aikana. [20]

Palomuurin toiminnallisuuden on vastattava sille asetettua tarvetta, minkä takia tulee tarkastella, mikä on sopivin palomuurityyppi liikenteen rajoittamiseksi verkkojen välillä [18]. Palomuurityypit voidaan luokitella kolmeen eri kategoriaan: pakettisuodattimet (traditional packet filters), yhteyssuodattimet (stateful packet filters) ja sovellustason yhdyskäytävät (application gateway). Pakettisuodattimet tarkastelevat jokaista lähtevää ja tulevaa pakettia erikseen ja päättävät käyttäjän asettamien suodatussääntöjen perusteella, tuleeko paketti sallia vai kieltää. Suodatussääntöinä voidaan käyttää esimerkiksi lähettäjän tai vastaanottajan IP-osoitetta, protokollatyyppiä tai TCP/UDP kohde- tai lähtöporttia.

Yhteyssuodattimet tarkastelevat myös paketteja, minkä lisäksi ne käyttävät TCP/IP-yhteyden tilatietoja, joiden perusteella liikennettä arvioidaan ja valvotaan koko istunnon ajan. Yhteyssuodattimet luokitellaan rajoittaviksi palomuureiksi, sillä niihin voidaan pakettisuodatussääntöjen lisäksi konfiguroida asetus, jonka perusteella suodattimet kieltävät kaikki paketit, mikäli paketit eivät kuulu hyväksyttyihin istuntoihin.

Pakettisuodattimet eivät pysty tarkkailemaan tai havaitsemaan sovellustasolla olevaa epätoivottua liikennettä, minkä takia sovellustason yhdyskäytäviä tulee käyttää pakettisuodattimien lisäksi. Sovellustason yhdyskäytävät ovat palvelimia, joiden kautta sovelluksen lähettämä ja vastaanottama data välitetään. Palvelimet tarkkailevat dataa ja tekevät sen perusteella päätöksen sallitaanko vai estetäänkö liikenne. Sovellustason yhdyskäytäviä voidaan käyttää esimerkiksi rajaamaan käyttäjäryhmien pääsyä tiettyihin palveluihin. [20]

Palomuurien ja DMZ:n käytön yhteydessä tulee ohjelmistot ja sääntökannat säännöllisesti päivittää, minkä lisäksi aliverkkojen välisen tietoliikenteen ja järjestelmälokitietojen tarkkailu on ehdottoman tärkeää. Laiminlyönnin seurauksena tunkeutuminen saattaa jäädä havaitsematta tai palomuurin haavoittuvuus korjaamatta. [18]

6.2.4 Järjestelmien koventaminen

Järjestelmien koventamisen tarkoituksena on poistaa käytöstä tarpeettomat järjestelmän toiminnallisuudet rajoittamalla järjestelmäkokoonpanoa tai -asetuksia. Tällöin tarpeettomat toiminnallisuudet ja niiden haavoittuvuudet eivät lisää riskitekijöitä. Etenkin Windows PC-työasemat ja palvelimet vaativat koventamista käyttöjärjestelmän laajoista ominaisuuksista ja automaatiojärjestelmän kannalta tarpeettomista, aktiivisista toiminnallisuuksista johtuen.

Laitetasolla automaatiojärjestelmistä tulee poistaa tai rajoittaa kaikki ylimääräiset liitännät ja rajapinnat, joiden avulla järjestelmään on mahdollista tunkeutua tai ladata haittaohjelmia. Järjestelmän konfigurointi ja ohjelmalliset muutokset tulee suojata, jotta ainoastaan järjestelmävalvojan on mahdollista muuttaa kokoonpanoa. Järjestelmän verkkoyhteydet rajoitetaan vain sallittuihin kohteisiin, jolloin laite voi lähettää ja vastaanottaa viestejä vain laitteilta, joiden MAC-osoitteet on sallittu. Lisäksi osassa järjestelmistä käytetään heartbeat-signalointia, jolla tarkkailleen ja ylläpidetään laitteen toimintatilaa. Mikäli järjestelmässä käytetään kyseistä signalointia on sen tarvitsema tietoliikenne dokumentoitava ja mahdolliset haavoittuvuudet arvioitava.

Ohjelmistotasolla järjestelmästä poistetaan kaikki tarpeettomat sovellukset ja palvelut. Ohjelmistoasennukset tulee suorittaa kontrolloidusti, jotta järjestelmään ei asenneta tietoturvattomia ohjelmistoja. Käytössä olevia ohjelmistoja tulee säännöllisesti seurata ja kehittää, jotta toiminnalliset viat ja haavoittuvuudet saadaan korjattua. Järjestelmästä poistetut tai rajoitetut sovellukset ja palvelut sekä uudet ohjelmistoasennukset ja ohjelmistokorjaukset tulee dokumentoida. [18]

6.2.5 Hyökkäysten ja haittaohjelmien tunnistaminen

Verkkoliikenteen ja isäntäkoneiden tapahtumien seurantajärjestelmien (Intrusion Detection System, IDS) avulla voidaan tunnistaa tunkeutumisyritykset tai muu verkkoon kuulumaton tietoliikenne. Verkkoliikenteen perusteella saatetaan tunnistaa myös viallinen tai väärin konfiguroitu laite. Useimmat seurantajärjestelmät käyttävät erityyppisiä tunnistusmetodeja, joiden perusteella ne analysoivat verkkoliikenteen ja sovellusten aktiivisuutta. Yleisimmät tunnistusmetodit luokitellaan kolmeen eri luokkaan [21]:

- Signature-perusteinen tunnistusjärjestelmä tarkkailee tapahtumia ja vertaa niitä tietokantansa sisältöön tunnistaakseen haitallisen tapahtuman. Signaturet ovat opetettuja malleja tai kaavoja, jonka perusteella tunnistaminen tehdään, minkä se tunnistaa tehokkaasti tunnetut uhat, mutta ei tuntemattomia uhkia. Lisäksi ne eivät pysty käsittelemään useita ja monimutkaisia tapahtumia, minkä takia seurantajärjestelmä saattaa pettää suuressa kuormituksessa.
- Anomalia-perusteinen tunnistusjärjestelmä käyttää profiileja, joihin on luokiteltu normaalin toiminnan ominaispiirteet. Tunnistusjärjestelmä tarkkailee aktiivista toimintaa ja vertaa sitä profiilien kynnysarvoihin. Mikäli kynnysarvo ylittyy, luokitellaan tapahtuma epäilyttäväksi. Anomalia-perusteiset tunnistusjärjestelmät eivät perustu mihinkään nykyiseen, tunnettuun tietoon, minkä takia ne voivat tunnistaa myös uusia, tunnistamattomia uhkia. Profiilien tekeminen on kuitenkin vaikeaa ja ne eivät usein kuvaa oikeaa tilannetta riittävän hyvin, minkä takia uhka jää tunnistamatta tai järjestelmän käyttö aiheuttaa vääriä hälytyksiä.
- Protokolla-analyysia käyttävät tunnistusjärjestelmät tarkkailevat protokollien tilaa ja toimintaa. Valmiiksi määritettyjen profiileihin vertaamalla järjestelmä päättää onko protokollan toiminta normaalia vai ei. Muihin järjestelmiin verrattuna protokolla-analyysilla pystytään tunnistamaan monimutkaisemmat hyökkäykset. Protokolla-analyysin käyttö on kuitenkin erittäin vaikeaa ja resursseja kuluttavaa, eikä sen perusteella voida tunnistaa hyökkäystä, mikäli hyökkäyksessä käytetyn protokollan toiminta on normaalia.

Tapahtumien seurantajärjestelmät ovat suhteellisen kevyitä tapoja valvoa verkon liikennettä, minkä takia ne ovat yleistymässä myös automaatiopuolella. Automaatioverkon työpisteiden seurantajärjestelmät sen sijaan saattavat aiheuttaa yllättäviä viiveitä, mistä aiheutuu toiminnallisia haittoja. Automaatiosovellusten toiminta saatetaan myös tulkita epäilyttäväksi ja on vaarana, että sovellus laitetaan karanteeniin, jolloin sen käyttö es-

tyy. Hyökkäysten ja haittaohjelmien tunnistamisjärjestelmien toiminta tuleekin varmistaa kaikissa olosuhteissa, etteivät ne estä suojattavan järjestelmän käyttöä tai muuten vahingoita sitä. [18]

6.3 Tietoturvan arviointi

Tietoturva-arvioinnin tavoitteena on selvittää järjestelmän tietoturvallisuus vertaamalla arvioinnin kohteen ominaisuuksia sille asetettuihin vaatimuksiin. Arvioinnin perusongelmana on vahva tapauskohtainen riippuvuus, minkä takia yksityiskohtaista ja yleispätevää toteuttamistapaa ei ole. Tässä luvussa sekä luvussa 6.4 - tietoturvan testausmenetelmät on pääosin käytetty lähteenä Pasi Ahosen kirjoittamaa TITAN-käsikirjaa ([18]), jossa on yhdistetty alan standardeja ja parhaita käytäntöjä.

6.3.1 Arviointikohteen määrittely

Arviointikohteen määrittelyssä rajataan, mitä järjestelmän ominaisuuksia ja osia tai osa-alueita arvioinnissa tarkastellaan. Arviointikohteet kartoitetaan järjestelmän kokonaisriskiarvioinnin yhteydessä, missä dokumentoidaan osat, joiden toteutuksessa ja toiminnassa on havaittu puutteita ja haavoittuvuuksia. Myös järjestelmän toiminnan kannalta erittäin kriittiset osat on syytä dokumentoida ja mahdollisesti arvioida, vaikka niissä ei välittömästi puutteita havaittukaan. Riskiarvioinnin ja tietoturvatavoitteiden perusteella päätetään, mitä arviointikohteita halutaan tutkia tarkemmin.

Yksittäisissä arvioinneissa kohteen ominaisuudet rajataan tarkasti, koska pienempien kokonaisuuksien tarkastelulla päästään yleensä tarkempaan lopputulokseen. Voidaan esimerkiksi tarkastella hyökkäyksen vaikutusta järjestelmään tai lähdekoodin tietoturvallisuutta. Määrittelyn tuloksina saadaan arviointikohteen ja tutkittavien ominaisuuksien rajaus.

6.3.2 Arviointikriteeristön määrittely

Arviointikriteeristön määrittelyssä kuvataan toimintaympäristön tietoturvamäärittelyt, -säännökset ja -vaatimukset sekä arviointikohteelle asetetut tietoturvavaatimukset, joita vasten arviointikohteen tietoturvallisuus todennetaan. Kriteeristö on loppujen lopuksi organisaation itsensä määrittelemä, mutta joitakin yleistyksiä sen suhteen voidaan tehdä.

Toimintaympäristön vaatimukset määräävät hyvin pitkälti kriteeristön referenssitason ja yksityiskohdat, minkä takia jokaisella arvioitavalla osalla tulisi olla määritettynä käytössä oleva tietoturvataso (security level). Tietoturvatasoon katsotaan kuuluvan tietoturvakontrollit, sekä tekninen että hallinnollinen tietoturvapolitiikka. Lisäksi järjestelmän suojattavien kohteiden vyöhykkeet ja vyöhykkeiden tietoturvatasot tulee olla määriteltynä järjestelmän arkkitehtikuvauksessa.

Standardien ja parhaiden käytäntöjen pohjalta automaatiojärjestelmien suojaamiseksi asetetaan vaatimuksia, joita organisaatio tarkentaa ja vertaa käytössä oleviin tietoturvavaatimuksiin ja -käytäntöihin. Seuraavat vaatimukset ovat hyvin yleisiä:

- toiminnan turvaaminen ja jatkuvuus, suunnitelmat vikatilanteiden varalle
- saatavuuden ja käytön varmistaminen, resurssien käytön kontrollointi
- järjestelmän suojauksen ja eristämisen kontrollointi
- pääsynvalvonnan ja käyttäjätilien kontrollointi
- prosessinohjauskyvyn ja valvontatiedon saannon turvaaminen
- järjestelmien kovennus ja rajapintojen palveluiden rajoittaminen
- tallennetun datan ja tiedonsiirron eheyden suojaaminen
- turvajärjestelmien varmistaminen
- tapahtumien jäljitettävyys
- haittaohjelmasuojauksen kontrollointi.

6.3.3 Arviointimenetelmien ja työkalujen määrittäminen

Arviointimenetelmiä ja työkalujen valitsemisvaiheessa määritetään yksityiskohtaisesti menetelmiä ja työkalut arvioinnin suorittamiseksi, sekä muun muassa mitä työkalujen asetuksia, haavoittuvuusprofiileita, tarkistuslistoja ja laajennuksia käytetään haluttuun tavoitetilaan pääsemiseksi. Arviointimenetelminä voidaan käyttää:

- haastatteluita
- haavoittuvuusanalyysia
- hyökkäysten sietokykyä testaavia menetelmiä
- järjestelmän asetusten selvittäminen ja vertaaminen määriteltyyn.

Lista on vain suuntaa-antava, menetelmiä sekä työkaluja on hyvin paljon erilaisiin evaluointitarkoituksiin. Valitun menetelmien ja työkalujen joukon tulee sisältää erilaisia lähestymistapoja järjestelmän ominaisuuksien todentamiseksi. Osa työkaluista saattaa kuitenkin vaatia erittäin syvällistä asiantuntemusta ja vuosien perehtyneisyyttä ja pohjatyötä. Luvussa 6.4 - Tietoturvan testausmenetelmät käydään tarkemmin läpi joitakin näistä menetelmistä. Työkaluja kyseisten menetelmien toteuttamiseen ei kuitenkaan käsitellä tässä työssä.

6.3.4 Arvioinnin suorittaminen ja raportointi

Tässä vaiheessa varsinainen arviointi eli testaus suoritetaan. Testauksessa käytetään ennalta määrättyjä menetelmiä ja työkaluja. Kuten mainittu, testausympäristön rakentaminen on suuritöistä, minkä takia jo tuotannossa olevien automaatiojärjestelmien testaus on käytännössä mahdotonta.

Uuden järjestelmän käyttöönottovaiheessa voidaan kuitenkin testata tietoturvaan liittyvät toiminnot, jolloin toimintaympäristö on hyvin lähellä todellista. Testausvaiheisiin kuuluu tarkastaa että:

- tietoturvaan liittyvät asetukset ja ominaisuuksien asennukset on suoritettu oikein
- määritelmän mukaiset tietoturvakontrollit toimivat oikein
- tietoturvapolitiikat ja -vaatimukset toteutuvat

- turvallisuustoiminnot ja kokonaisjärjestelmä toimivat oikein.

Testauksen tuloksena saadaan tyypillisesti tiedostoja, joihin testausmenetelmien ja työkalujen käytöstä kerätyt tiedot ja data tallennetaan.

Evaluoinnin tulisi olla helposti toistettavissa, jolloin evaluointikokonaisuudelle tulisi määritellä realistinen perustaso, jota sovelletaan tapauskohtaisesti. Toistettavuuden kannalta myös evaluointiraportit ovat ehdottomia. Raporteista tulisi käydä ilmi evaluointikokonaisuuden tunnistetiedot, kriteeristö, metodit ja työkalut, tulokset ja parannusehdotukset ja kehityskohteet.

6.3.5 Arviointitulosten todentaminen

Arviointitulosten todentamisessa analysoidaan tulosten oikeellisuutta ja tarkkuutta. Todentaminen monesti edellyttää taustatietoa edellisistä evaluointituloksista ja toimintaympäristöstä, minkä takia organisaatiot käyttävät usein asiantuntija-apua. Lisäksi evaluointitulosten määrä voi olla valtava, minkä takia tulisi määritellä menettelytapa, jolla tuloksia voidaan todentaa tehokkaasti.

6.4 Tietoturvan testausmenetelmät

Toimistoverkon ja automaatioverkon tietoturvan testaamiseen on kehitetty joukko testityökaluja, joiden sopivuus on jälleen kerran tapauskohtaista. Toimistoverkon laitteiden ja ohjelmistojen testaamiseen tyypillisesti käytetään seuraavia työkaluja:

- verkon rakenteen ja palveluiden kartoitukseen käytetyt työkalut
- penetraatio-testerit
- robustisuus-testerit ja palvelunesto-testerit
- haavoittuvuusskannerit
- sovellus-testerit
- lähdekoodianalysaattorit.

Automaatioverkon laitteiden ja ohjelmistojen testauksessa käytetään lisäksi laitekohtaisten konfiguraatioiden tarkistusmenetelmiä, automaatiospesifisiä testereitä ja suorituskyvyn monitorointityökaluja. Automaatioverkkojen tapauksessa ei tosin käytetä sovellustestereita ja lähdekoodianalysaattorin käyttö on myös harvinaisempaa, koska sen vaikutus ohjelmistokehitysprosessin jälkeiseen valmiiseen ohjelmistokoodiin on pieni.

6.4.1 Verkon rakenteen ja palveluiden kartoitus

Verkon rakenteen ja palveluiden kartoitukseen käytetään usein porttiskannausta ja verkkotiedustelua. Näillä työkaluilla voidaan tunnistaa verkon tietoturvaloukkaukset, haavoittuvat palvelut ja poikkeamat sallituista palveluista ja kerätä näistä todistusaineistoa. Lisäksi kokonaisvaltaisella verkkoskannauksella saadaan hyvä lähtökohta penetraa-

tiotestaukselle ja samalla skannauksen tiedot auttavat tunkeutumisen havaitsemisjärjestelmän konfiguroinnissa.

Verkkotiedustelu voidaan suorittaa aktiivisella tiedustelulla, jolloin verkon kohteeseen lähetetään dataa ja tulevasta datasta tehdään johtopäätökset. Verkon liikennettä voi myös vain kuunnella, jolloin puhutaan passiivisesta tiedustelusta. Tiedustelulla pyritään tunnistamaan verkossa olevat laitteet ja palvelut.

Porttiskannaus on hyvin systemaattista ja järjestyksessä jokaiselle kohteen portille lähetetään yhteyspyyntö. Yhteyspyyntöön vastanneista porteista voidaan tehdä johtopäätöksiä, kuten mihin tarkoitukseen porttia käytetään tai onko portti avoin. Vastauksen perusteella skannauksen tehnyt henkilö voi myös päätellä kohteen porttikohtaiset sovellukset tai käyttöjärjestelmän tulkitsemalla tuloksia. Porttiskannaus voi haitata verkon toimintaa, koska se käyttää kaistaa ja hidastaa verkkoa. Tämä on otettava huomioon etenkin verkoissa, joiden sovellukset vaativat reaaliaikaisuutta. Porttiskannauksen on myös raportoitu aiheuttaneen yllättäviä seurauksia automaatioverkoissa, minkä takia työkalujen sopivuus tulee varmistaa.

6.4.2 Kestävyystestaus ja palvelunestotestaus

Kestävyystestaus keskittyy tahattomien vikatilanteiden etsintään. Testauksen avulla voidaan varmistua siitä, että kohde on riittävän kestävä täyttääkseen sille kohdistetut tietoturvavaatimukset. Tahattomien vikatilanteiden testaus on tärkeää, koska vikatilanteita voidaan hyödyntää hyökkäyksessä tai niiden seurauksena paljastuu luottamuksellista tietoa. [17]

Palvelunestotestauksessa simuloidaan palvelunestohyökkäyksiä ja analysoidaan testauskohteen toiminnan jatkuvuutta. Palvelunestohyökkäystapoja on kaksi, joista ensimmäisen tavan tarkoituksena on generoida mahdollisimman paljon liikennettä kohdeverkkoon, jolloin verkon resurssit loppuvat ja verkko ruuhkautuu käyttökelvottomaksi. Toinen tapa on hyväksi käyttää esimerkiksi verkon reitittimen tai päätelaitteen haavoittuvuuksia ja tehdä laitteen palvelusta toimintakyvytön. [22]

6.4.3 Haavoittuvuusskannaus

Haavoittuvuusskannaus keskittyy yleisessä tiedossa olevien haavoittuvuuksien löytämiseen. Haavoittuvuuksia ovat muun muassa vaaralliset ohjelmointivirheet, avonaiset palvelut tai vanhat ohjelmistoversiot, jotka ovat osoittautuneet turvattomiksi. Tunnetut haavoittuvuudet on tallennettuna skannerin tietokantoihin, minkä takia tietokannat on pidettävät jatkuvasti ajan tasalla, jotta tulokset ovat luotettavia.

Teollisuusautomaatiossa etenkin vanhat ohjelmistoversiot ovat hyvin yleisiä, minkä takia haavoittuvuusskannaus on tärkeä testausmenetelmä. Skannerit eivät pysty kuitenkaan arvioimaan verkon kokonaisriskitilannetta vaan tulokset viittaavat yksittäisiin riskeihin. Myös haavoittuvuusskannauksen kohdalla on syytä noudattaa suurta huolellisuutta, eikä tuotannossa olevia järjestelmiä tule testata. Testausta varten järjestelmä tai järjestelmän osa tulee eriyttää tuotannosta.

6.4.4 Penetraatiotestaus

Penetraatiotestauksessa kohteena olevaa järjestelmää vastaan hyökätään joukolla tunnettuja hyökkäystapoja, jotka toimivat testeinä. Testaustavat voidaan luokitella black box -testeihin ja white box -testeihin. Black box -testeissä testaajalle ei anneta mitään tietoa testattavasta järjestelmästä tai infrastruktuurista. Black box -testaus siis simuloi järjestelmän haavoittuvuutta ulkopuolista hyökkääjää vastaan. White box -testauksessa testaajalla on kaikki olennainen tieto järjestelmästä tai infrastruktuurista, jolloin testaus simuloi sisäisen hyökkäyksen tai tietovuodon seurausta. Tietovuodon seurauksena hyökkääjällä voi olla esimerkiksi järjestelmän käyttöoikeudet.

Tunkeutuminen voidaan suorittaa hyökkäyksellä, jolla pyritään läpäisemään kohdejärjestelmän palomuurit. Tällöin puhutaan suorasta tunkeutumistestauksesta. Epäsuorassa tunkeutumistestauksessa pyritään aiheuttamaan vikatilanteita, joiden aikana kohdejärjestelmään yritetään tunkeutua. [17]

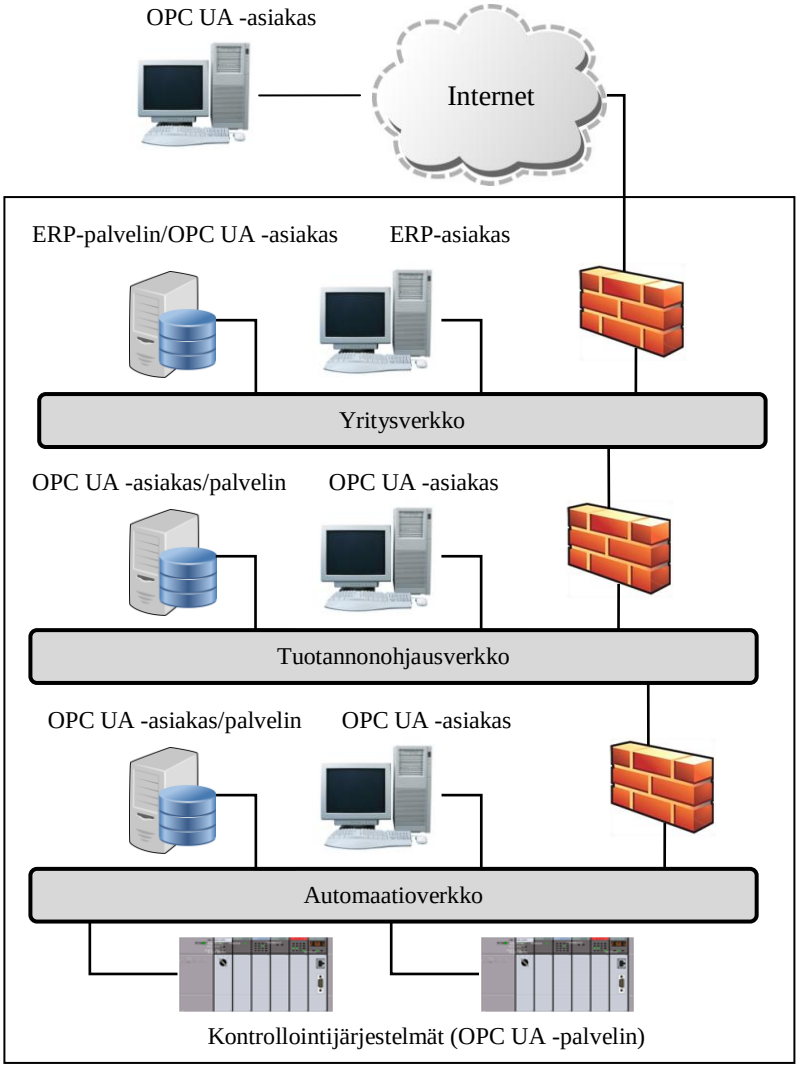
Testauksen aikana järjestelmän toimintaa tulee monitoroida ja analysoida aktiivisesti, jonka jälkeen kaikki löytyneet haavoittuvuudet käydään läpi. Haavoittuvuuksien vaikutukset arvioidaan ja tarkastellaan mahdollisia korjaustapoja. Penetraatiotestaus on tehokas testausmenetelmä ja toimii samalla auditointimenetelmänä organisaatiolle. Etenkin black box - testaus edellyttää kuitenkin asiantuntijuutta ja huolellista suunnittelua, koska huolimaton testaus voi aiheuttaa suurta vahinkoa kohdejärjestelmille.

7 OPC UA:N TIETOTURVA

Tietoturvallisuuden merkitys on huomattu myös OPC Foundationin keskuudessa ja tietoturvaa on parannettu huomattavasti OPC UA:ssa. Käyttöön on otettu julkisen avaimen järjestelmä ja varmenteet, joilla voidaan tehokkaasti suojata kahden osapuolen välinen tiedonsiirtokanava ja istunto.

7.1 Tietoturvamallin rakenne

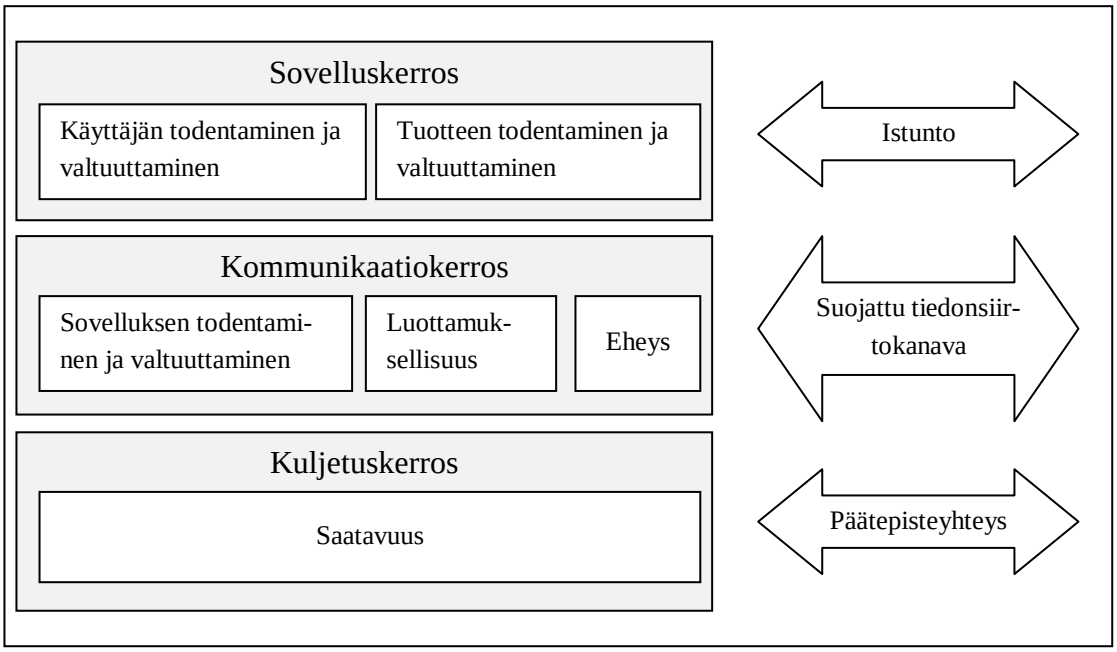
Tietoturvamallin rakenteessa on huomioitu, että OPC UA -sovelluksia käytetään erilaisissa toimintaympäristöissä, joiden tietoturvavaatimukset vaihtelevat. Kuvassa 18 on esitetty esimerkki mahdollisesta toimintaympäristöstä.



Kuva 18: OPC UA:n toimintaympäristö

Kuvan mukaisessa toimintaympäristössä joudutaan tekemään kompromisseja muun muassa tietoturvallisuuden ja tehokkuuden välillä. Automaatioverkkojen osalta tehokkuus on tärkeämpi kriteeri kuin tietoturva, koska dataa pitää pystyä siirtämään nopeasti ja tehokkaasti, jotta tuotantoprosesseja voidaan kontrolloida. Automaatioverkon tietoturva on kuitenkin hyvin kriittinen tekijä, jolloin tietoturva-aspektia ei voida kokonaan sivuuttaa. Vastaavasti yritysverkoissa tietoturva on tärkeämpää, koska verkot ovat yhteydessä Internetiin, eivätkä sovellukset ole yhtä reaaliaikakriittisiä kuin automaatioverkkojen sovellukset. Edellä mainitun takia tietoturvamallin tulee olla joustava, jotta OPC UA -sovelluksia voitaisiin käyttää automaatiojärjestelmien kokonaisvaltaiseen yhdistämiseen.

Tietoturvamalli rakentuu useammasta kerroksesta, joista jokaisella on oma tietoturvallisuuteen liittyvä tehtävänsä. Kuvassa 19 on esitetty yhteyden toisen osapuolen tietomallin rakenne, kerroksien tehtävät sekä kerroksien välillä muodostettavat yhteydet. Yhteyden molempien puolien tietomallin rakenteet ovat samanlaiset.



Kuva 19: Kerroksittainen tietoturvamalli ja kerroksien tehtävät [16]

Sovelluskerroksen tehtävänä on muodostetun istunnon aikana lähettää automaatiolaitteilta saatua dataa OPC UA -asiakkaan ja palvelimen välillä. Istuntoa käytetään käyttäjän ja tuotteen todentamiseen sekä valtuuttamiseen.

Kommunikaatiokerrosten välillä luodaan suojattu tiedonsiirtokanava, joka suojaa yhteyden osapuolten välistä tiedonsiirtoa. Sovellusten todentamiseen ja valtuuttamiseen käytetään suojatun tiedonsiirtokanavan palveluita sekä konseptia, jonka ideana on, että sovellukset voivat tunnistaa muita sovelluksia. Sovellusten lähettämien viestien eheyden ja luottamuksellisuuden tarkistamiseen käytetään digitaalisia allekirjoituksia sekä viestin sisällön salaamista.

Suojattua dataa lähetetään ja vastaanotetaan kuljetuskerrosten välisen päätepisteyhteyden kautta. Virheistä toipumismekanismien käyttö on siirtokerroksella pakollista, jotta esimerkiksi palvelunestohyökkäykset eivät estä datan saatavuutta.

7.2 Julkisen avaimen järjestelmä

Tässä luvussa esitellään julkisen avaimen järjestelmä (Public Key Infrastructure), koska OPC UA:n tietoturvatoteutukset pohjautuvat varmenteisiin, joiden hallinnointiin käytetään kyseistä järjestelmää. Julkisen avaimen järjestelmä sisältää itsenäisiä kokonaisuuksia, joilla on erityiset tehtävät, jotta varmenteilla voidaan todentaa niille osoitettuja asioita. Itsenäisiin kokonaisuuksiin kuuluu viranomaistahoja, jotka hallitsevat kirjauksia (Registration Authority), varmenteita (Certification Authority) ja varmenteiden hyväksyntää (Validation Authority). Lisäksi on varmenteen loppukäyttäjä (End-Entity). Entiteetit voivat tapauskohtaisesti olla esimerkiksi sovelluksia, sovelluksen käyttäjiä tai laitteita.

Kirjauksia hallinnoiva viranomaistaho käsittelee loppukäyttäjän lähettämiä varmenteita koskevia kysymyksiä ja ohjaa kysymykset varmenteita hallinnoivalle viranomaistaholle. Varmenteita hallinnoiva taho myöntää, uudistaa ja kumoaa varmenteita. Kirjauksia ja varmenteita hallitsevat tahot on yleensä yhdistetty. Varmenteiden oikeellisuus tarkistetaan varmenteen hyväksyntää hallinnoivalla taholla.

7.3 Varmenteet

Varmenteet ovat elekronisessa muodossa olevia kolmannen osapuolen vahvistamia dokumentteja. Varmenteita käytetään julkisen avaimen järjestelmässä julkisten avainten jakamiseen loppukäyttäjien välillä siten, että varmenteen saaja voi olla varma toisen osapuolen identiteetistä. Samalla varmistetaan, että julkista avainta tai siihen liittyvää dataa ei ole voitu muuttaa.

OPC UA:ssa käytetään kolmea erilaista varmennetta, jotka ovat kaikki X.509v3 -tyypin varmenteita: sovellusvarmenne (Application Instance Certificate), ohjelmistovarmenne (Software Certificate) ja käyttäjävarmenne (User Certificate). Sovellusvarmenne identifioi isäntäkoneella käynnissä olevan OPC UA -sovelluksen instanssin. Sovellusvarmenne on pakollinen kaikissa OPC UA -tuotteissa. Ohjelmistovarmenne puolestaan identifioi OPC UA -tuotteen version. Varmenne sisältää lisäksi tiedon, mitä profiileja kyseinen versio tukee, jotta molemmat osapuolet tietävät onko kommunikointi mahdollista. Käyttäjävarmenteilla identifioidaan käyttäjä, joka yrittää muodostaa yhteyden palvelimelle. OPC UA sisältää muitakin tapoja käyttäjävaltuuksien tunnistamiseksi, joten käyttäjävarmenne ei ole pakollinen.

7.4 Tiedonsiirtokanavan suojaus

Asiakkaan ja palvelimen välinen tiedonsiirto suojataan kahdessa vaiheessa. Ensimmäisessä vaiheessa suojataan tiedonsiirtokanava ja toisessa vaiheessa sovellusten välinen istunto.

7.4.1 Suojattu tiedonsiirtokanava

Kun asiakas tietää palvelimen haun päätepisteen, se lähettää suojaamattoman pyynnön, johon palvelin vastaa lähemmällä tiedot istunnon päätepisteestä ja tietoturva-asetuksista. Tietoturva-asetuksiin kuuluu muun muassa tietoturvakäytännöt (Security Policies), tietoturvatilat (Security Modes), käyttäjän valtuuksien menettelytavat (User Token Policies) ja palvelimen sovellusvarmenne. Tiedot saatuaan asiakas valitsee tietoturva-asetuksiltaan itselleen sopivan istunnon päätepisteen. Samalla asiakas vahvistaa palvelimen lähettämän varmenteen hyväksynnän antaneen viranomaistahon (Validation Authority) allekirjoituksen perusteella. Palvelimelle lähetetään suojattu pyyntö suojatun tiedonsiirtokanavan muodostamisesta, mikäli sertifikaatti on todettu luotettavaksi.

Pyyntö voidaan suojata eri tietoturvatiloilla: allekirjoituksella (Sign), allekirjoituksella ja salauksella (SignAndEncrypt) tai jättää kokonaan suojaamatta (None). Allekirjoitukseen asiakas käyttää oman sovellussertifioinnin yksityisavainta (Private Key). Viestin salaamiseen asiakas käyttää palvelimen sovellussertifioinnin julkista avainta (Public Key).

Pyynnön saatuaan palvelin poistaa salauksen sertifikaatin osalta ja vahvistaa varmenteen oikeaksi viranomaistahon allekirjoituksen perusteella. Mikäli asiakkaan sovellussertifikaatti todetaan luotettavaksi, palvelin avaa salauksen omalla yksityisavaimellaan ja todentaa allekirjoituksen asiakkaan julkisella-avaimella. Palvelin lähettää vastauksen, joka on suojattu vastaavalla tavalla ja asiakas tekee viestille vastaavat tarkistusmenettelyt. Tällöin molempien osapuolten sovellusten todentaminen ja valtuuksien tarkastaminen on suoritettu onnistuneesti.

Asymmetristen avainten käyttö vaatii kuitenkin prosessointitehoa, minkä takia niitä käytetään pääasiassa asiakkaan ja palvelimen välisen jaetun salaisuuden (shared secret) vaihtoon. Jaettua salaisuutta käytetään asiakkaan ja palvelimen välisen yhteisen, symmetrisen avaimen muodostamiseen. Asiakas ja palvelin käyttävät symmetristä avainta jatkossa viestien allekirjoittamiseen ja salaukseen.

Muodostettu suojattu tiedonsiirtokanava tulee kuitenkin uusia aika-ajoin, jotta suojausta ei pystytä purkamaan pitemmänkään hyökkäyksen aikana. Tällöin menettely asiakkaan ja palvelimen välillä täytyy käydä uudelleen alusta alkaen, jotta symmetriset avaimet saadaan muodostettua.

7.4.2 Istunto

Istunnon muodostamiseen käytetään tiedonsiirtokanavan yhteydessä sovittuja tietoturva-asetuksia sekä johdettua symmetristä avainta. Asiakas lähettää salatun pyynnön istun-

non luomiseksi, johon palvelin vastaa lähettämällä ohjelmistovarmenteen (Software Certificate), jolla palvelin osoittaa toiminnalliset ominaisuutensa sekä todistaa omistavansa varmenteen, jota käytettiin tiedonsiirtokanavan muodostamisen yhteydessä. Ohjelmistosertifikaatit ovat sertifiointia hallitsevan viranomaistahon (Certification Authority) ylläpitämiä, ja ne osoittavat, että tuote tai tuotteen versio on varmennettu. Ohjelmistovarmenteella siis todennetaan tuote ja varmistetaan tuotteen valtuudet.

Mikäli tietoturvatiloina käytetään allekirjoitusta tai allekirjoitusta ja salausta, asiakas lähettää pyynnön yhteydessä myös haasteen (nonce, challenge), jonka tehtävänä on todentaa palvelin oikeaksi. Haasteeseen palvelimen on vastattava allekirjoittamalla vastaus omalla yksityisavaimella. Tämän jälkeen asiakas validioi ohjelmistovarmenteen ja tarkastaa läpäisikö palvelin haasteen.

Istunto pitää aktivoida ennen kuin sitä voidaan käyttää. Asiakas lähettää aktivointipyynnön (ActivateSession), joka sisältää asiakkaan ohjelmistosertifikaatin sekä käyttäjätiedot. Ohjelmistosertifikaatin oikeellisuus varmennetaan viranomaistaholla, mutta käyttäjätietojen tarkistamiseen on olemassa useampia ratkaisuita, jotka riippuvat tietojen esitystavasta. Käyttäjän todentamiseen ja valtuuksien varmistamiseen voidaan esimerkiksi käyttää käyttäjätietokantaa, jossa verrataan käyttäjätunnus-salasana -paria tai jälleen kerran viranomaistahoa, mikäli valtuudet ovat sertifikaattimuodossa. Kun istunnon aktivointi on tehty ja kaikki varmennukset on suoritettu onnistuneesta, yhteys on täysin muodostettu.

8 KOHDEYRITYKSEN TIETOMALLI

Työn alkuvaiheessa määritettiin tietomallia koskevat rajaukset. Kuvassa 2 kuvatun ISA-95:n määrittämän toiminnallisen hierarkiamallin mukaisesti taso 4 eli liiketoiminta-suunnittelu ja logistiikka rajattiin työn ulkopuolelle. On kuitenkin syytä mainita, että kokonaisvaltaisella järjestelmäintegraatiolla saavutetaan suurimmat laadulliset ja taloudelliset hyödyt. Tietomallissa siis huomioidaan tuotannon hallintaan liittyvät toiminnot, valvonta ja kontrollointijärjestelmät, sekä sensorit ja toimilaitteet ja näiden väliset tietovirrat.

Kohdeyrityksen tietomallin suunnittelussa käytettiin luvussa 2 mainittuja ISA-standardeja, sekä MESA:n tarjoamaa dokumentaatiota parhaista käytännöistä. Mallit ovat alkuperäisessä muodossaan varsin monimutkaisia ja raskaita, eikä MESA suosittele niitä käytettäväksi sellaisinaan kuin hyvin suurten yritysten yhteydessä, minkä takia malleja muokattiin kohdeyritykselle sopiviksi. Tietomallin suunnittelun aikana kävi ilmi, että kohdeyrityksen nykyiset tietorakenteet on suunniteltu vastaavien standardien pohjalta, minkä takia tiedonkeruumallin sovittaminen onnistui luontevasti.

8.1 Haastattelut

Tietomallin suunnittelua ja yleisesti tiedonkeruuhanketta varten haastateltiin kohdeyrityksen tuotanto- ja laatupäälliköitä. Haastatteluiden perusteella oli tarkoitus saada muodostettua kokonaiskuva tuotannonohjauksesta, vaatimuksista, sekä ongelmakohdista. Painotus oli kuitenkin siinä, millaisia ongelmia ja tarpeita tiedonkeruuhankkeella voitaisiin ratkaista.

Tuotantopäälliköiden haastatteluista kävi ilmi seuraavia ongelmakohtia:

- Nykyisiä tietojärjestelmiä on liikaa ja osa järjestelmistä on päällekkäisiä. Järjestelmien välillä on tiedonsiirto-ongelmia, minkä takia tietoa ei pystytä hyödyntämään, eikä tiedon oikeellisuudesta ja luotettavuudesta ole varmuutta. Lisäksi yksittäisistä tehdasohjelmista tulisi päästä eroon, minkä nähtiin olevan tärkeä tekijä tuotantojärjestelmien yhtenäistämisessä.
- Raportit eivät ole yhdenmukaisia ja selkeitä, minkä takia niiden käyttö on kankeaa. Raportit eivät myöskään palvele niiden käyttötarkoitusta.
- Tuotantoympäristössä tietoa syötetään edelleen paljon manuaalisesti, mikä aiheuttaa ongelmia tiedon luotettavuuden ja oikeellisuuden kannalta, koska inhimillisiä erehdyksiä ja unohtamisia tapahtuu satunnaisesti. Manuaalinen tiedon syöttö aiheuttaa myös tarkastuskierteen, koska tieto pitää tarkastaa toisen henkilön toimesta.

Tuotantopäälliköt antoivat myös seuraavia kehitysideoita:

- Yleisesti tuotannon käytäntöjä tulisi yhtenäistää tietojärjestelmien kanssa sillä tällä hetkellä tietojärjestelmät eivät tue käytäntöjä riittävän hyvin.
- Automaatiolaitteille tulisi saada reaaliaikaiset hälytysjärjestelmät, jotka ilmoittaisivat mikäli mittauspisteen raja-arvot ylittyvät. Tällöin asiaan voitaisiin puuttua välittömästi eikä jälkikäteen. Reagointi jälkikäteen aiheuttaa myös tarkastuskierteen, koska tuotteille joudutaan tekemään tarkastusmenettelyjä tai tuotteet joudutaan hylkäämään.
- Tuotannon työpisteille sähköinen kirjautuminen esimerkiksi kulkukortilla ja sähköiset työmääräimet, jolloin vältytään ylimääräisiltä papereilta.
- Työpisteille paneelit, joista työntekijät näkisivät tuotantoajon kokonaisuudessaan. Tämän nähtiin helpottavan työntekijöiden omien aikataulujen suunnittelua esimerkiksi ruokatauon kannalta. Lisäksi välitön palautejärjestelmä suoritetusta työstä saattaisi motivoida työntekijöitä.
- Tuotantolinjakohtaisesti tarkemmat häiriömittarit. Tarkemmat häiriömittarit auttaisivat muun muassa investoinneissa, kun tiedettäisiin tarkalleen mistä laitteesta tuotantolinjan pysähtyminen johtui. Tällä hetkellä häiriömittarit antavat vain linjatason tiedon eivätkä laitekohtaista tietoa.

Laatupäälliköiden haastatteluissa esiin tuli seuraavia asioita:

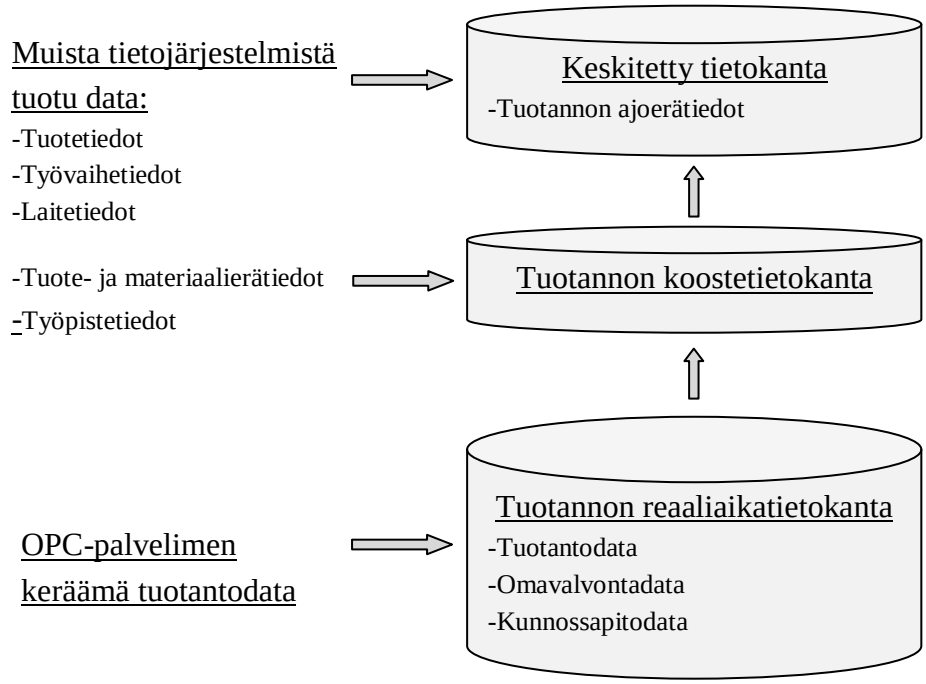
- Laadunvalvonnan kannalta on kriittisiä hallintapisteitä, joista kerätty tieto on ehdottoman tärkeää. Kriittisistä hallintapisteistä tulisi saada dataa automaattisesti. Tällä hetkellä datan eteen joudutaan tekemään paljon manuaalista työtä, joka vaikuttaa datan oikeellisuuteen ja luotettavuuteen.
- Datasta johdettua tietoa tallennetaan edelleenkin paljon paperisessa muodossa, minkä takia arkistointi ja jäljitettävyys on vaikeaa.
- Kerättyä dataa ei pystytä hyödyntämään, koska tietoa ei viedä tai sitä ei saada vietyä oikeassa muodossa tietojärjestelmiin.
- Tuotteiden jäljitettävyyden parantamiseksi halutaan yksikäsitteinen tunnus, jonka kautta nähdään koko tuotantoprosessi, jonka läpi tuote on kulkenut.

8.2 Saatavilla oleva data ja datan tallennus

Tuotantolaitteista kerättävä data voidaan karkeasti jakaa kolmeen eri luokkaan: tuotantodata, omavalvontadata ja kunnossapitodata. Tuotantodataan kuuluu kaikki tuotannon mittaamista varten kerättävä data ja omavalvontaan kaikki tuotteiden laatuun vaikuttava data. Kunnossapitodataan kuuluu muun muassa hälytykset, säätöarvojen muutokset ja laitevikailmoitukset.

Tuotantoympäristössä laitteilta kerättävää dataa saattaa tallentua jopa useita satoja gigatavuja päivässä. Datan valtavasta määrästä johtuen tuotannosta kerätty data tallennetaan reaaliaikatietokantaan, josta data tietyin väliajoin kootaan koostetietokantaan. Koostetietokantaan siirretään koostettua dataa, kuten keskiarvoja tai minimi- ja maksimiarvoja datasta, jolloin datan määrä pienenee murto-osaan alkuperäisestä. Täl-

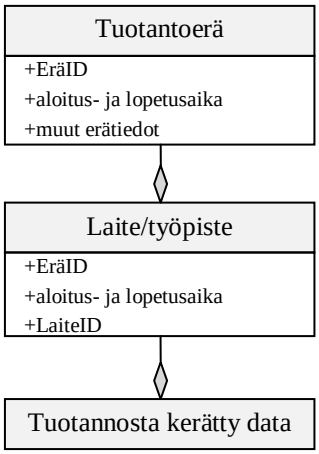
löin ensimmäinen tietokannan dataa ei tarvitse säilyttää kovin pitkiä aikoja vaan puhutaan muutamasta vuorokaudesta, jolloin tarkalla prosessidatalla on vielä merkitystä. Alla olevassa kuvassa 20 on esitetty kyseinen korkean tason tietomalli.



Kuva 20: Tuotannon ajoerätietojen koostaminen keskitettyyn tietokantaan.

Kuvan 20 mukaisesti reaaliaikainen data yhdistetään tuote- ja materiaalierätietoihin tuotannon koostetietokannassa. Koostetietokannan data yhdistetään keskitetyssä tietokannassa tuotetietoihin, työvaihetietoihin ja laitetietoihin, jolloin saadaan muodostettua tuotannon ajoerätiedot kokonaisuudessaan.

Alemman tason tietomallissa tuotantodata, omavalvontadata ja kunnossapitodata luokiteltiin yksityiskohtaisemmin. Tässä mallissa tuotannosta kerätty data yhdistetään työpisteeseen tai laitteeseen, mitkä puolestaan yhdistetään tuotantoerään, jolloin päästään tilanteeseen, jossa eräkäsite identifioi yksiselitteisesti koko tuotantoprosessiketjun, jonka läpi yksittäinen tuote on kulkenut. Tässä mallissa kuitenkin ajauduttiin ongelmaan, joka käsitellään luvussa 8.3. Kuvassa 21 on vielä yksinkertaistettu kuvaus tästä tietomallista.



Kuva 21: Matalan tason tietomallikuvaus.

Kerätty data on yhdistetty laitteeseen tai työpisteeseen laiteID:lla ja vastaavasti työpisteet on yhdistetty tuotantoerään eräID:lla. Lisäksi tunnistuksessa käytetään ajon aloitus- ja lopetusaikoja. Tuotannon datatauluilla on lisäksi oma sisäinen ID, joka identifioi taulun sisäisesti yksittäisen data-alkion. Tietomalleista tehtiin kohdeyritykselle yksityiskohtaisempi dokumentointi, jossa tietomallit on kuvattu ja luokkien sekä attribuuttien merkitykset selitetty.

8.3 Ongelmat tietomallin suunnittelussa

Tietomallin suunnittelussa ajauduttiin ongelmaan erätiedon kanssa. Erätieto siis tulisi tuoda ulkopuolisesta tietojärjestelmästä tuotannon koostetietokantaan. Ongelmaksi muodostui, saadaanko eräkäsite yhdistettyä työpisteeseen tai laitteeseen yksiselitteisesti. Riittävän yksiselitteinen ratkaisu saatiin vasta työn viime hetkillä. Tietomallikuvaukseen lisättiin ajon aloitus- ja lopetusajat, jolloin tietyllä aikavälillä laitteet ja työpisteet pystytään sitomaan eräkäsitteeseen. Eri tuotteiden tapauskohtaisista tuotantoprosessiketjuista johtuen voi kuitenkin tulla tilanteita, ettei ajon aloitus- ja lopetusajat riitä yksinään vaan tarvitaan suurempia muutoksia.

Tuotannon tietojärjestelmien tietovirroista ei ollut dokumentaatiota. Järjestelmien lukumäärästä johtuen ei myöskään saatu täyttä varmuutta siitä, sopivatko muut tietojärjestelmät rajapinnoiltaan sellaisinaan tiedonkeruuhankkeeseen. Ainakin erätiedon takia joudutaan vielä tekemään tarkennuksia ja mahdollisesti järjestelmämuutoksia yksittäiseen tietojärjestelmään.

8.4 Suositukset tietomallin suunnittelussa esiintyneiden asioiden pohjalta

Kohdeyrityksen tulee jatkossa ehdottomasti panostaa dokumentointiin. Mitä enemmän tietojärjestelmiä on käytössä ja mitä laajempia integrointiratkaisuja suunnitellaan, sitä suurempiin ongelmiin ajaudutaan, mikäli dokumentointia ei tehdä. Lisäksi nykyisten tietojärjestelmien väliset tietovirrat tulisi testata ja dokumentoida, jotta pohja tiedonkeruuhankkeelle olisi mahdollisimman hyvin tunnettu.

Tietojärjestelmien suuri määrä aiheuttaa tälläkin hetkellä ongelmia, minkä takia tiedon käytettävyys on huono. Osa järjestelmistä nähtiin päällekkäisiksi ja koettiin, etteivät ne palvele käyttötarkoitusta kovinkaan hyvin. Tämän takia ennen tiedonkeruuhankkeen aloittamista tulisi miettiä, ovatko kaikki järjestelmät välttämättömiä ja voitaisiinko järjestelmien toiminnallisuuksia yhdistää. Ennen tiedonkeruuhanketta järjestelmämuutokset on todennäköisesti yksinkertaisempi tehdä kuin järjestelmäintegraation jälkeen. Lisäksi pienempi järjestelmämäärä helpottaa niiden hallintaa.

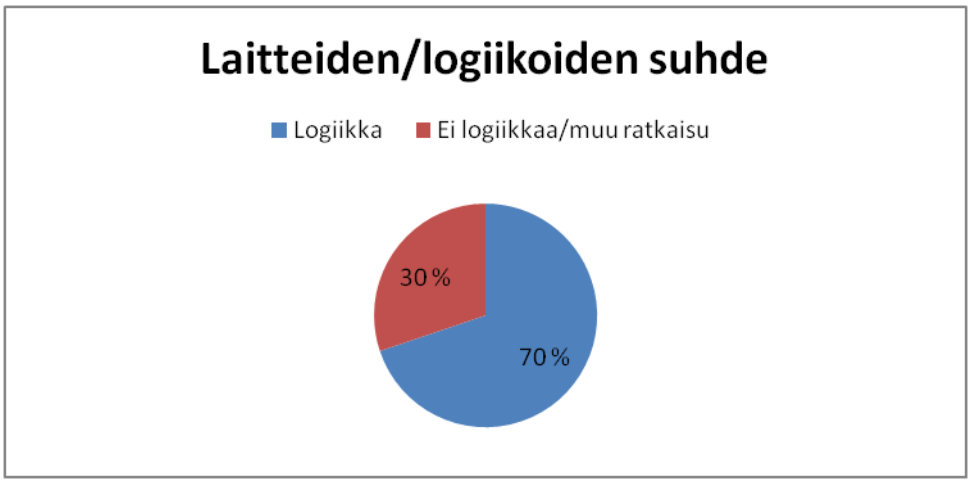
9 LAITEKARTOITUKSEN AIKANA ILMENNEET HAVAINNOT JA TULOKSET

Laitekartoituksena tarkoituksena oli dokumentoida tuotannossa olevat tuotantolaitteet. Dokumentissa kiinnitettiin huomiota tuotantolaitteiden ohjausperiaatteisiin ja tiedonkeruun kannalta oleellisiin liitäntöihin. Samalla kirjattiin verkkolaitteet ja -osoitteet, mikäli laite oli jo yhdistetty Ethernet-verkon kautta johonkin tietojärjestelmään.

Ennen varsinaista laitekartoitusta käytiin tutustumassa tuotantoympäristöön ja prosesseihin yleiskuvan muodostamiseksi. Tuotantolaitteiden dokumentointi jouduttiin tekemään kahteen kertaan, koska ensimmäisellä kerralla ei osattu kiinnittää huomiota oleellisiin asioihin ja kartoitus jäi puutteelliseksi. Lopullinen dokumentti antaa suhteellisen hyvän kokonaiskuvan tuotannon laitejärjestelmistä.

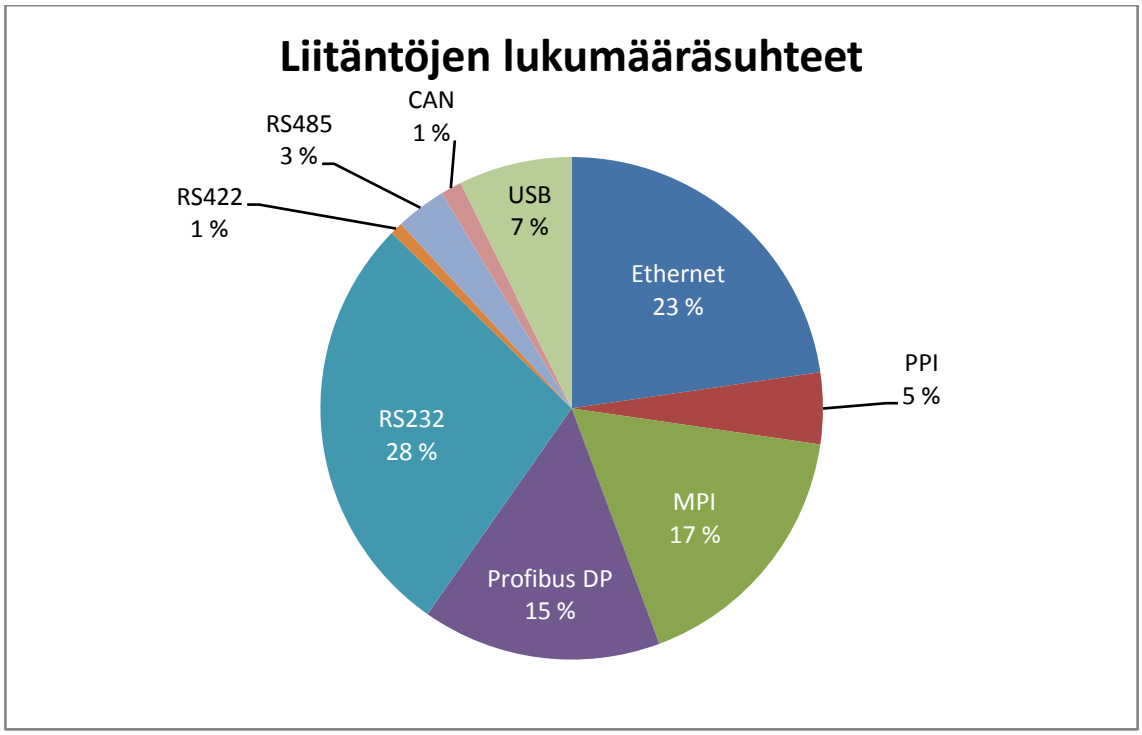
9.1 Laitekartoituksen kokonaiskuva

Laitekartoituksen aikana kirjattuja tuotantolaitteita on yli 400 kappaletta. Näistä suurin osa käytti Siemensin tai Omronin valmistamia logiikoita. Yleisiä olivat myös automaatiojärjestelmissä käytettävät ohjauskortit sekä teollisuustietokoneet. Osa laitteista ei ollut helposti avattavissa, jotta tuotannon aikana olisi voitu tarkastaa laitteen ohjausperiaate. Todennäköistä kuitenkin on, että edellä mainitut laitteet ovat suljettuja kokonaisuuksia, jotka sisältävät valmistajan oman ohjauskortin. Kaikki laitteet eivät myöskään sisältäneet mitään digitaalista ohjausjärjestelmää vaan ohjaus hoidetaan täysin analogisin säädöin. Kuvaajassa 1 on esitetty prosentuaalinen osuus laitteista, joiden ohjaus perustuu logiikkaan.



Kuvaaja 1: Tuotantolaitteiden ja logiikoiden suhde

Kuvaajassa 2 on esitetty laitteiden liitäntöjen lukumääräsuhteet. On huomioitava, että kuvaajan prosentuaaliset arvot ovat suuntaa antavia, koska osa laitteiden liitännöistä on tulkinnanvaraisia. Esimerkiksi RS485-standardia tukeva liitäntä tukee myös RS422-standardia. Myös MPI-liitäntä on sähköisesti RS485:ta vastaava, mutta Siemens luokittelee sen MPI-liitännäksi, koska logiikka käyttää MPI-protokollaa.



Kuvaaja 2: Liitäntöjen lukumääräsuhteet

Liitäntöjä on myös paneeleissa, kytkimissä, toistimissa ja modeemeissa. Näitä ei kuitenkaan kuvaajassa 2 huomioitu, koska on täysin tapauskohtaista voitaisiinko niitä hyödyntää tiedonkeruussa. Muun muassa hyvin monessa laitteessa logiikka on yhdistetty RS232-liitännän kautta paneeliin, jossa on Ethernet-liitäntä. On mahdollista, että paneeli pystyy toimimaan datan välittäjänä, mutta se on epävarmaa.

On myös mainittava, että laitekartoitus ei ole täydellinen. Kunnossapitotyöntekijät, jotka tuntevat laitteet parhaiten, osallistuivat laitekartoituksen tekemiseen, mutta työntekijöiden tietotaidolla ja asenteilla havaittiin olevan hyvin suuri vaikutus, koska kaikki laitekartoituksen aikana kerätty tieto pohjautuu heidän tietoihinsa. Dokumentista puuttuu laitteita pelkästään jo inhimillisen muistin takia. Toisaalta osa laitteista tuli dokumentoitua jopa kahteen kertaan, koska työntekijät antoivat laitteelle eri merkityksen ja ulkopuolisena henkilönä näitä on vaikea erottaa. Duplikaatit kuitenkin usein huomattiin vertailemalla laitteiden logiikkakokoonpanoa, sijaintia ja käyttötarkoitusta.

9.2 Yksittäisen tuotantolinjan pilotointi

Yksittäisen tuotantolinjan pilotoinnin tarkoituksena oli saada tietoa, mitä dataa tuotantolinjoista on mahdollista kerätä. Kyseinen linja on suhteellisen uusi ja se koostuu yhdestä

isommasta kokonaisuudesta, joka käsittää linjan alkupään, sekä seitsemästä yksittäisestä laitekokonaisuudesta. Linjassa on kaksi laitetta, joiden ohjausperiaatteesta ei ole varmuutta. Kuutta muuta laitetta ohjataan logiikoilla, joista neljään oli saatavilla kommentoitu ohjelma.

Tiedonkeruuhankkeen kannalta kiinnostavaa dataa on saatavilla kyseisestä tuotantolinjasta seuraavasti:

- omavalvontadata: ilman ja laiteosien lämpötilat, ilmankosteus, tuuletus- ja jäähdytystiedot, tuotereseptin ja asetusarvojen tarkistus
- tuotantodata: laitteiden ajotilat ja -nopeudet, läpimenoajat
- kunnossapitodata: laitteen osien ja moottoreiden sekä releiden vikatiedot, hälytykset, ohjelmavirheet, virtauksien ja venttiilien ohjausarvot.

Tuotantodatan kannalta oleellinen linjan tuotantomäärä on todennäköisesti myös saatavilla linjaan asennetuista valokennoista tai valoverhoista. Logiikoiden muistipaikoista kyseistä tietoa ei kuitenkaan löytynyt.

9.3 Haasteet ja ongelmakohdat

Suurin osa laitekartoituksen aikana ilmenneistä haasteista ja ongelmakohdista johtui siitä, ettei yhteisiä käytäntöjä tiedonhallinnasta ollut sovittu ja automaatiojärjestelmien hankinnasta ei ollut laadittu ohjeistusta. Edellä mainituista puutteista johtuen kokonaiskuva automaatiojärjestelmien hallinnasta oli epäselvä.

9.3.1 Automaatiojärjestelmiä koskeva tiedonhallinta yleisesti

Automaatiojärjestelmiä koskeva tiedonhallinta todettiin puutteelliseksi, eikä tiedonhallinnasta ollut sovittu yhteisiä käytäntöjä. Laitteita, järjestelmiä ja verkkoinfrastruktuuria ei ollut dokumentoitu tai dokumentointi oli puutteellista. Arkistointimenettelyt olivat yleisesti ottaen sekavia, eikä selkeää arkistointitapaa ollut. Logiikoiden varmuuskopioita ja valmiiksi verkkoon yhdistettyjen laitteiden dataa oli tallennettu satunnaisille verkkolevyille. Tästä kaikesta päätellen kunnossapidon ja tietohallinnan vastuualueiden raja oli epäselvä, eikä automaatiojärjestelmien tiedonhallintaa ollut osoitettu kenellekään.

9.3.2 Puutteelliset logiikoiden ohjelmat sekä versionhallinta

Laitekartoituksen yhteydessä oli tarkoitus tarkastella logiikoiden ohjelmia ja muistipaikoista löytyvää dataa, jotta olisi saatu parempi kuva, mitä dataa yksittäiseltä logiikalta on mahdollista kerätä. Tähän ei kuitenkaan päästy, koska logiikoiden ohjelmat olivat puutteellisia tai niistä ei ollut olemassa varmuuskopiota, jota olisi voitu tarkastella.

Hyvin suuressa osassa logiikoiden ohjelmista puuttui kommentit, jolloin muistipaikkojen datasisällöstä ei ole mitään tietoa. Oletettavasti laitetoimittajat ovat tällä tavoin halunneet salata ja suojata logiikat, jolloin asiakkaan ainut mahdollisuus on tilata kaikki työ laitetoimittajalta. Voi olla, ettei laitetoimittaja ole myöskään toimittanut oh-

jelman varmuuskopiota, mutta sekavasta arkistointimenettelystä johtuen on mahdollista, että varmuuskopio on yksinkertaisesti hävinnyt.

Yhtä logiikkaa kohden saattoi löytyä lukuisia ohjelmia. Ohjelmaversioita ei ollut kommentoitu yksiselitteisesti tai muuten dokumentoitu, minkä takia oli mahdotonta päätellä, mikä on laitteen nykyinen ohjelmaversio. Koska automaatiojärjestelmien elinkaari on hyvin pitkä, ohjelmiin on todennäköisesti tehty muutoksia ajan myötä.

Ohjelmien nimeäminen aiheutti myös ongelmia. Ohjelmien niminä saattoi olla projektinumeroita, logiikoiden mallisarjanimiä tai jopa pelkkiä päivämääriä, jolloin ohjelmaa on käsitelty. Ohjelmaa ja logiikkaa ei voitu näin ollen yhdistää varmasti toisiinsa. Lisäksi löytyi hyvin paljon ohjelmia, jotka olivat tyhjiä. Mitään tarkoitusperää näille ei keksitty.

9.3.3 Suljetut laitejärjestelmät

Tuotantokäytössä on myös paljon suljettuja laitejärjestelmiä, joiden toimintaan kukaan kohdeyrityksen henkilöistä ei pysty vaikuttamaan. Laitteiden ohjauksen muutos- ja korjaustyöt on pakko tilata laitetoimittajalta. Vikatilanteisiinkaan ei pystytä puuttumaan vaan laite on yhdistettävä etäyhteyden kautta laitetoimittajaan. Laitteiden ohjauskortit on myös tyypillisesti piilotettu laitteen sisälle, minkä takia laite olisi jouduttu purkamaan lähes kokonaan laitekartoituksen takia. Tähän ei kuitenkaan ryhdytty.

9.4 Suositukset laitekartoituksen aikana ilmenneiden asioiden pohjalta

Tehtyjen havaintojen ja ilmenneiden ongelmakohtien perusteella tarkasteltiin, mitä osa-alueita tulee parantaa ja mitä jatkossa tulee tehdä, jotta hanketta voidaan viedä eteenpäin. Kohdeyritykselle laadittiin myös dokumentti automaatiojärjestelmien hankinnassa huomioon otettavista asioista, jotta jatkossa huomioitaisiin tiedonkeruun kannalta oleelliset asiat ja vältyttäisiin virheiden toistamiselta.

9.4.1 Arkistointi ja dokumentointi

Ensimmäisenä tulee päättää kenen vastuualueella automaatiojärjestelmien tiedonhallinta on. Sekavasta arkistointitavasta, dokumenttien versionhallinnasta ja lukuisista tyhjistä tiedostoista johtuen tiedon käytettävyys on huono, mikä laskee tiedon arvoa. Arkistoinnista ja dokumentoinnista tulee sopia yhteiset käytännöt, joiden mukaan menetellään.

Dokumentointi laitteiden, järjestelmien ja verkoninfrastruktuurin osalta tulee saattaa ajan tasalle ja jatkossa päivittää muutoksien osalta. Kattava dokumentointi näiden osalta on lähtökohta tietoturvalliselle toimintaympäristölle ja helpottaa huomattavasti tulevaisuuden hankintaprojekteja.

9.4.2 Tuotantolinjakohtaiset projektit jatkossa

Tiedonkeruuhankkeen toteutuksessa tulee jokaisesta tuotantolinjasta tehdä omakohtainen projekti. Projektien pohjana voidaan hyödyntää laitekartoitusdokumenttia, mutta myös laitteiden tarkempaa tarkastelua varmasti vaaditaan. Projekteja suunniteltaessa kannattaa ensiksi tarkastella, kuinka hankala työ linjan liittäminen tulee olemaan ja aloittaa helpoimmista projekteista. Jo edellä mainitut puuttuvat logiikkaohjelmat aiheuttavat sen, että helpostakin projektista tulee äärimmäisen vaikea. Logiikoiden ohjelmat tulee siis pyytää laitetoimittajalta ennen kuin projektia on mielekästä aloittaa.

On myös varmasti tilanteita, jossa kaikkea dataa ei voida kerätä linjan kaikilta laitteilta tai se ei ole kannattavaa. Esimerkiksi jos yksittäisestä laitteesta ainut kerättävä data olisi käyntitieto niin sen takia laitetta tuskin kannattaa liittää verkkoon, mikäli sen eteen joudutaan tekemään paljon työtä. Projektin aloitusvaiheessa kannattaakin haastatella uudelleen tuotanto-, laatu- ja kunnossapitopäälliköitä siitä, mikä data on oikeasti tärkeää, jotta ei tehtäisi turhaa työtä.

Jo tämän diplomityön aikana tehtiin linjamuutoksia, jotka aiheuttivat pieniä sekaannuksia laitekartoitusvaiheessa. Linjakohtaiset muutokset ja siirrot tulee jatkossa päivittää laitekartoitusdokumenttiin, jolloin yhtä suurta kartoitusta ei tarvitse tehdä enää jatkossa.

9.4.3 Automaatiolaitehankinnat jatkossa

Automaatiolaitteiden hankintavaiheessa kunnossapidon tulee konsultoida tietohallintoa, jotta laitteeseen saadaan tiedonkeruun kannalta asianmukaiset logiikkakokoonpanot. Logiikkakokoonpanossa pitää olla erillinen kommunikaatioprosessori, jossa on Ethernet-liitäntä. Laitetoimittajilta tulee pyytää logiikan täydellinen, kommentoitu ohjelma, sekä ohjelman varmuuskopio. Lisäksi laitetoimittajalle ilmoitetaan logiikoiden halutut verkkoasetukset, sekä kerättävä data, jolle valmistaja tekee muistiosien logiikan ohjelmaan. Edellä mainituilla toimenpiteillä laitteet ovat jatkossa tiedonkeruun kannalta yhdenmukaisempia.

Suljettuja laitejärjestelmiä tulee välttää. Suljetut laitejärjestelmät ovat pelkästään kunnossapidon kannalta hankalia ja tiedonkeruuhankkeen kannalta jopa mahdottomia. Mikäli tällaisia järjestelmiä ei voida välttää, on laitetoimittajalta vaadittava ainakin edellä mainitut asiat.

Tekijänoikeuksien kannalta on yleisesti ottaen parasta ostaa logiikkaohjelman tekijänoikeudet järjestelmätoimituksen yhteydessä. Ohjelmat eivät saa olla salattuja tai muuten suojattuja järjestelmätoimittajan toimesta. Tällä vältetään toimittajariippuvuus jatkossa.

10 YHTEENVETO

Tuotannon tiedonkeruun standardointi ja kokonaisvaltaisen järjestelmäintegraation toteuttaminen on monimutkainen ja pitkä prosessi, johon tarvitaan hyvin paljon resursseja. Ennen hankkeen aloittamista tulee tehdä valtavasti taustatyötä, jotta hanketta on mielekästä viedä eteenpäin ja turhilta virheiltä ja kustannuksilta vältyttäisiin. Integraation ohjaamiseen ja toteuttamiseen on onneksi kehitetty viimeisten vuosien aikana standardoituja toimintamalleja ja teknologioita, joihin tässä työssä tutustuttiin. Järjestelmäintegraation yhteydessä aikaisemmin eristyksissä olleet automaatioverkot altistuvat tietoturvauhille, joiden torjumiseksi joudutaan tekemään kompromisseja tietoturvan ja käytettävyyden välillä. Tietoturvallisuutta käsiteltiin yleisesti, eivätkä vaatimukset, arvioinnit tai testausmenetelmät koske yksinomaan työn tilannutta yritystä.

Työn aikana kohdeyritykselle saatiin tiedonkeruuta varten luotua ISA-standardien ja MESA:n parhaiden käytäntöjen pohjalta standardin mukainen tietomalli. Tietomalliin ja yrityksen tietojärjestelmiin tarvitaan vielä muutoksia ennen kuin kokonaisuudesta saadaan yhteensopiva. Tähän vaiheeseen kannattaa erityisesti käyttää resursseja, koska hyvin suunnitelluilla tietomalleilla ja tietokannoilla taataan se, että datan hyödynnettävyys on mahdollisimman korkea. Samalla varmistetaan, ettei kerätystä datasta muodostu rasitetta tietojärjestelmille.

Kohdeyrityksen tuotantolinjat dokumentointiin tiedonkeruun kannalta oleellisin osin ja tarkasteltiin mahdollisia liitäntävaihtoehtoja tiedonsiirron kannalta. Dokumentti antaa suhteellisen hyvät lähtötiedot tuotantolinjojen nykytilanteesta ja se kannattaa pitää jatkossa ajan tasalla. Sekä tietomallin suunnittelun, että tuotantolinjojen laitekartoituksen aikana tehtiin paljon havaintoja ja huomattiin suuria ongelmakohtia, jotka voivat olla jopa esteenä hankkeen etenemiselle, ellei niitä saada korjattua. Puutteellinen dokumentointi ja yhteisten käytäntöjen puuttuminen olivat suurimmat ongelmien aiheuttajat ja näihin asioihin kohdeyrityksen tulee ehdottomasti panostaa jatkossa.

Kuten edellä on mainittu, järjestelmäintegraation toteuttaminen vaatii valtavasti resursseja ja tästä syystä monet suuret teolliset tuotantoyritykset ovat pitkittäneet hankkeen aloittamista. Tuotannon taloudelliset ja laadulliset vaatimukset kasvavat kuitenkin jatkuvasti, jolloin järjestelmäintegraation toteuttaminen tulee todennäköisesti olemaan hyvin monen teollisen tuotantoyrityksen edessä tulevaisuudessa. Vaikka integraatiosta saadut taloudelliset ja laadulliset hyödyt eivät ilmene välittömästi, on niiden merkitys niin suuri, että integraatio vaikuttanee jatkossa yritysten kilpailukykyyn.

LÄHTEET

[1] Scholten, B. The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing. 2007, ISA.

[2] MESA. Business To Manufacturing Markup Language, Equipment, Version 6.0. 2013.

[3] MESA. Business To Manufacturing Markup Language, Material, Version 6.0. 2013.

[4] MESA. Business To Manufacturing Markup Language, Personnel, Version 6.0. 2013.

[5] MESA. Business To Manufacturing Markup Language, Process Segment, Version 6.0. 2013.

[6] MESA. Business To Manufacturing Markup Language, Batch Production Record, Version 6.0. 2013.

[7] Component Object Model. Microsoft. [viitattu 2.1.2015]. Saatavissa: http://msdn.microsoft.com/en-us/library/windows/desktop/ms680573(v=vs.85).aspx.

[8] Distributed Component Object Model. Microsoft. [viitattu 2.1.2015]. Saatavissa: http://technet.microsoft.com/en-us/library/cc958799.aspx.

[9] Stripf, W. PROFIBUS: Open Solutions for the World of Automation. Julkaisussa: Integration Technologies for Industrial Automated Systems. 2006.

[10] Swales, A. Open Modbus/TCP Specification. 1999, Schneider Electric.

[11] MODICON, Inc., Industrial Automation Systems. Modicon Modbus Protocol Reference Guide. 1996.

[12] Siemens. Communication with SIMATIC System Manual. 2006.

[13] CAN in Automation (CiA). CAN History. [viitattu 3.1.2015]. Saatavissa: http://www.can-cia.de/index.php?id=systemdesign-can-history.

[14] USB Implementers Forum, Inc. Universal Serial Bus 3.1 Specification, Revision 1.0. 2013.

[15] Lange, J. Iwanitz, F. OPC - Openness, Productivity, and Connectivity. Julkaisussa: Integration Technologies for Industrial Automated Systems. 2006.

[16] Mahnke, W., Damm, M., Leitner, S. &. OPC Unified Architecture. 2009, Springer-Verlag, Berlin Heidelberg.

[17] Teollisuusautomaation tietoturva, verkottumisen riskit ja niiden hallinta. 2010, Suomen Automaatioseura ry.

[18] Ahonen, P. TITAN-käsikirja, VTT:n päätuloksia Tekesin Turvallisuusohjelman TITAN-projektissa. 2010, VTT.

[19] Naedele, M. IT Security for Automation Systems. Julkaisussa: Integration Technologies for Industrial Automated Systems. 2006.

[20] Kurose, J., Ross, K. Computer Networking: A Top-down Approach. 5. painos. 2009, Pearson.

[21] Scarfone, K., Mell, P. Guide to intrusion detection and prevention systems (idps). Julkaisu. National Institute of Standards and Technology. 2007.

[22] Hussain, A., Schwab, S., Thomas, R., Fahmy, S., Mirkovic, J. DDOS experiment methodology. Proceedings of the DETER Community Workshop on Cyber Security Experimentation. 2006.